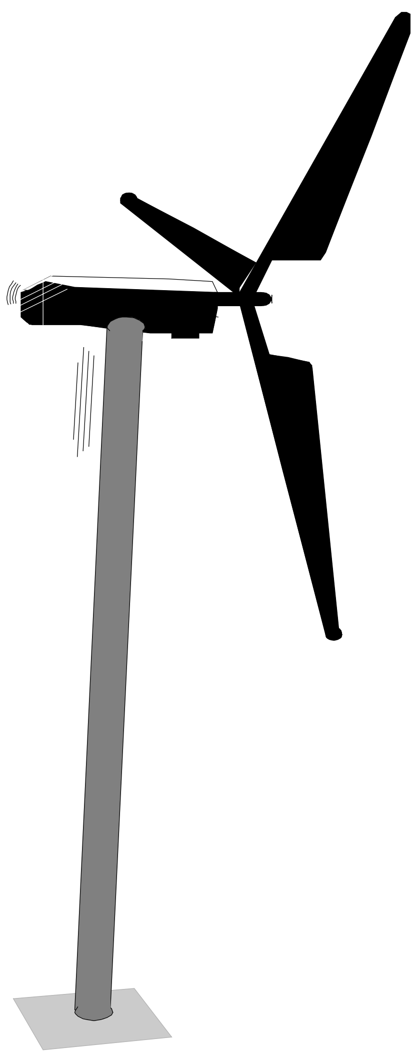
USER'S GUIDE

to the Wind Turbine Dynamics Computer Programs

YawDyn and AeroDyn for ADAMS®



A. Craig Hansen
David J. Laino

Mechanical Engineering Department
University of Utah
Salt Lake City, UT 84112

Phone: 801-581-4145
Fax: 801-581-8692
email: hansen@me.mech.utah.edu
dlaino@stress.mech.utah.edu

Program date and version
YawDyn 11.0, August, 1998

Prepared for the
National Renewable Energy Laboratory
under Subcontract No.
XAF-4-14076-02

Notice

Table of Contents

List of Figures

List of Tables

USER'S GUIDE

to the Wind Turbine Dynamics Computer Programs

YawDyn and AeroDyn for ADAMS®

About this Guide

This User's Guide is written to assist engineers with the preparation and use of two computer programs for wind turbine aerodynamics and dynamics analyses, 1) YawDyn and 2) AeroDyn for ADAMS®. These programs are vastly different in their capabilities for representing structural degrees of freedom. YawDyn has at most four degrees of freedom while ADAMS has virtually unlimited dof with 200-400 being typical for full turbine system models (but 2 or 3 being possible for simple models). In spite of this difference in the dynamics model, the aerodynamics models are identical. Both use the same subroutines to calculate aerodynamic forces on the blades and they use many of the same input data files. This makes it possible to write a User's Guide for both programs without placing an undue burden on those who wish only to use YawDyn.

This Guide has two major sections, with several chapters in each. The first section discusses all aspects of using YawDyn--installing the software, preparing input files, and program execution. Users who wish to use only YawDyn can obtain all the required information from this section. The second section provides the additional information required to use AeroDyn with ADAMS. This section assumes the reader is familiar with ADAMS and the documentation that is supplied with ADAMS. This Guide focuses only on the aerodynamics analysis for ADAMS. Readers should consult the ADAMS and ADAMS WT manuals for information on creating the structural dynamics model in ADAMS. Readers interested in using AeroDyn must be familiar with both the YawDyn and AeroDyn sections of this Guide. Anyone wishing to use AeroDyn is strongly encouraged to use YawDyn first. This will help familiarize them with the aerodynamics analysis without the added complication that is introduced by ADAMS. This advantage is gained at little cost, as all the data files used in YawDyn will also be required for AeroDyn when it is time to add the structural complexity to the model.

SECTION A: User's Guide to YawDyn

1.0 Major changes since the last version of YawDyn

This chapter is provided to assist experienced users of YawDyn or AeroDyn with the task of updating their models for use with the latest version of the code. New users can skip to the Introduction.

Version 11.0 of the codes contains a number of significant modifications that have been implemented since the last full release (version 10.0). These changes pertain to both the functionality of the codes and the user interface. The yawdyn.ipt file is not compatible with previous versions. You can use the Windows interface, YawDynVB, to translate your version 9.6 and 10.0 files into the version 11.0 format. We believe the new format will be much easier to use in the long run, though we regret that this change requires experienced users to modify their old data files. No changes were made to the airfoil data files, so the airfoil files are backwards compatible.

The hub-height wind file format has changed so these files are not compatible with previous versions. The new files have an extra column to permit simultaneous power-law and linear vertical shears. While this combination was possible in previous versions, it was accomplished using a confusing wind shear flag, which changed the definition of the last column of the hub-height wind files. This wind shear flag has been eliminated from yawdyn.ipt as explained below. This modification means the wind file columns do not change, regardless of the values of input flags. Of course, users can still change the source code to redefine the meanings of the columns if they wish (for instructions on how to accomplish such changes to the wind file format, see chapter 7.0.) The columns for a standard hub-height wind file in order are: 1) Time, 2) hub-height wind speed, 3) wind direction, 4) vertical wind speed, 5) linear horizontal shear coefficient, 6) vertical power law shear coefficient, 7) linear vertical shear coefficient, and 8) gust velocity. The first six columns are the same as in version 10.0. Users of the IECWind and WindMaker programs will need to download new copies of these programs as well to generate IEC wind files in the new format.

We have modified the YawDynVB Windows application introduced with version 10.0. YawDynVB will read yawdyn.ipt files in either the version 9.6, version 10.0, or version 11.0 formats. Although YawDynVB can also write yawdyn.ipt in version 10.0 format, this option should be used with caution as the resulting file may be missing parameters that were eliminated in version 11.0 of YawDyn. We hope users of YawDynVB will find some of our improvements to this code helpful. Many of the changes were necessary to reflect changes in yawdyn.ipt for version 11.0. Other functional additions include the ability to browse for both hub-height and turbulence files, and the ability to use formulas in the blade element data table.

The YawDyn and AeroDyn program files have been overhauled to use Fortran 90 formats and intrinsic functions. If you do not use a Fortran 90 compiler, we do not recommend you update from version 10.0 unless you plan to reprogram the code to Fortran 77 yourself. The program now uses binary full-field turbulence files instead of the fomatted type used in YawDyn 10.0. (To create these files you must update to SNLWIND-3D version 2.04 or later.) A new source code file, modules.f, has been added to the program to handle the FF variables. MODULES.FOR should be compiled to create two module files, FF_Wind.mod and Outputs.mod, that are required to create YawDyn and AeroDyn executables.

A major change has been made in the functionality of the dynamic inflow calculations of YawDyn, which are conducted when the DYNIN option is selected in yawdyn.ipt. The new dynamic inflow model is based on a modified Pitt and Peters model, which is a completely different theory than blade element/momentum (BEM) used for the EQUIL option. This is a significant change from the old DYNIN option of applying a first-order lag to the BEM induction factor. This new dynamic inflow model does not iterate for induction factor, as does the BEM method. When using DYNIN, YawDyn seeks a trim solution for the dynamic inflow parameters using the induction factor tolerance (ATOLER). If a trim solution is not found after 50 iterations, the simulation will terminate. The user should attempt a larger tolerance, or use the EQUIL option instead of DYNIN to run the simulation.

The dynamic inflow model (DYNIN option) significantly reduces simulation times compared to the blade element/momentum (BEM) model (EQUIL option), since it does not iterate for the induction factor as does the BEM model. The dynamic inflow effect is insignificant, and therefore results should be comparable to those from the BEM model, except when rapid changes in blade angle-of-attack occur. We recommend the use of the DYNIN option, but caution that it is not yet extensively tested, so review your results carefully for obvious errors.

Several other minor changes have been made in an effort to make the program less susceptible to "nuisance" errors. Unused parameters in the yawdyn.ipt file should no longer trigger errors. The existence of wind and other input files is checked, thus providing more useful error messages. The program conducts a more robust search for hub-height (HH) wind and full-field (FF) turbulence files. If YawDyn detects a different wind file type than is specified, a warning is issued that the file type specifier may be incorrect, and the program will attempt to use the other wind file type.

Changes to the appearance of the yawdyn.ipt file include both the removal and addition of several input parameters, and a couple of variable name changes. While reading the following description, the reader may find it useful to view the format of the new file in Table 6.1. As mentioned above, the shear flag (PWR or IEC) has been eliminated. Both a linear and power law vertical shear coefficient can now be included in hub-height wind files for all simulations. Users who do not use YawDyn VB to update their yawdyn.ipt files can simply delete the line containing this shear flag from yawdyn.ipt, but be sure to add the extra column in the wind files as mentioned earlier. Steady wind files (single line wind files) used with version 10.0 can be converted for use with version 11.0 using YawDynVB for YawDyn 11.0.

A simulation mode flag has been added on line 2 of the yawdyn.ipt file. When set to INTERACT, simulations run as they always have before. Selecting BATCH mode allows the programs the run unattended by replacing pause statements (that require user response) with a time delay function. Implementing this mode causes a simulation to resume automatically in the same manner it would if the user had opted to continue after a pause in interactive mode. One caveat to note in BATCH mode arises when running with the EQUIL option: If the induction factor calculation does not converge after 1000 iterations, the program will continue by setting the default value of zero for the induction factor. This situation will not arise when using the DYNIN option. The BATCH option was added to prevent the program from "hanging" on pause statements during unattended operation.

The element print parameters used to control the creation and contents of the element.plt (aelement.plt for ADAMS) file in version 10.0 have been eliminated from yawdyn.ipt. Instead, there is now a PRINT/NOPRINT flag at the end of each line in the blade element data table. This change allows for more precise control of which elements are output to the element.plt file. To print out data for a particular element the string 'PRINT' should be placed after the fifth parameter - the airfoil data file specifier (NFOIL) - for that element. If the program does not find the string 'PRINT', or finds the string 'NOPRINT' on an element data line, then data will not be written for that element. Users who do not wish to print out any element data can simply delete the element print parameter line from their version 10.0 files, and the element.plt file will not be created.

The rotor radius has also been eliminated from the yawdyn.ipt file. YawDyn and ADAMS now calculate the rotor radius (R) based on hub radius (R_{hub}), precone angle (PC) and blade length (L) using the equation:

$$R = R_{hub} + L * \cos(PC)$$

The blade length is calculated based on the location of the center (RELM) and the length (DR) of the outboard blade element. The value for the radius is still given in both YawDynVB and the file yawdyn.opt as calculated using this equation.

The definition of the FREE/FIXED option describing the simulated yaw condition has changed slightly. The FIXED flag now describes the *yaw rate* condition rather than the yaw condition. It works in conjunction with the initial yaw rate parameter in yawdyn.ipt. To describe a fixed yaw condition, use FIXED and <u>set the yaw rate to zero</u>. A non-zero yaw rate using FIXED simulates a constant yaw rate over the duration of the simulation. This is a useful feature for those who wish to model a constant rate yaw drive, but it also requires extra care when attempting to model a fixed yaw condition.

Also added in version 11.0 is the parameter PsiInit to set the starting position of the rotor in a YawDyn simulation (not used by ADAMS). This has been added immediately following RPM in yawdyn.ipt. This value follows the YawDyn convention where zero degrees indicates blade 1 down. Specifying a value of 0.0 (degrees) for PsiInit places the rotor in the same starting position as in all previous versions of YawDyn. This parameter was added mainly to facilitate direct comparison with other wind turbine dynamic analysis codes, which use other conventions.

The CNTRL/NOCNTRL option has been changed to SINGLE/MULTI, to reflect its purpose in a more general sense. SINGLE is used when a single airfoil table (or single operating point between two tables) is desired for an entire simulation. MULTI is used when a user-provided algorithm will cause a move between airfoil tables during a simulation. The maximum number of airfoil tables is set by the parameter MAXTABLE (replacing the old MAXPHI) in AERODYN.INC. The following parameter in the input file, previously called AILPHI(1) , is now named TableID. It is used to define the operating point when multiple airfoil tables are present in an airfoil data file. Users whose airfoil data files only contain a single airfoil table need not be concerned with these parameters; they are ignored unless more than 1 airfoil table is present in your airfoil files.

Another addition to yawdyn.ipt is a list of output parameters at the end of the file. This list defines the output channels for the yawdyn.plt file. This list allows many more options for output including parameters for other blades (besides just blade 1), the choice of units (e.g., kN-m or N-m), and the ability to define the order of the output columns in yawdyn.plt. A list of all available parameters is given in Table 6.3 of this Guide, as well as at the bottom of the sample yawdyn.ipt file distributed with YawDyn version 11.0, and on the Output tab in YawDynVB. When opening an old version of yawdyn.ipt in YawDynVB, a default list of outputs is generated to produce a yawdyn.plt file identical to previous versions of YawDyn. The user must include the desired channels in this list. If no channels are specified, the simulation will not run.

All output files created by YawDyn 11.0 now include a reference to the program and version number on the top line of the file. Users should bear this in mind when post-processing the results.

Detailed notes regarding the changes implemented in each version of YawDyn are shown in the comment statements at the beginning of the YAWDYN.FOR file. These comments may help if you are updating from an older version of the program.

The FOILCheck utility program has also been updated. The method of the program has not changed. Improvements focused mainly on making the program easier to use and the prompts easier to understand. A discussion of this program is presented in Appendix C of this Guide.

2.0 Introduction

This document is intended to provide information necessary to prepare inputs for the computer program YawDyn and the AeroDyn subroutines for use with ADAMS. The routines have a great deal in common, including most of the contents of the input data files. Section A of this guide contains information required to run YawDyn or ADAMS. Section B pertains only to the AeroDyn routines for ADAMS and can be ignored by anyone not planning to use ADAMS.

YawDyn and AeroDyn were developed and are maintained with the support of the National Renewable Energy Laboratory (NREL) National Wind Technology Center. YawDyn simulates the yaw motions or loads of a horizontal axis wind turbine with a rigid or teetering hub and two or more blades. The rotor can be simulated in steady winds, discrete time-series winds, or full-field, three-dimensional turbulent wind fields. This document provides a detailed description of each of the program inputs and operating instructions. Sample input and output files are provided for testing the program operation. There is no discussion of the underlying theory or limitations of the models. That discussion is available in a technical report and journal articles [see list of references].

In 1992, the aerodynamics analysis subroutines from YawDyn were modified for use with the ADAMS® program, which is available from Mechanical Dynamics, Inc. (Ann Arbor, MI). The YawDyn and ADAMS input data files are compatible with one another (and generally are identical). ADAMS allows the engineer to consider a model with virtually unlimited structural degrees of freedom and the AeroDyn subroutines provide the same aerodynamics methods that are used in YawDyn. This creates the most versatile and powerful wind turbine dynamics modeling capability known to the author.

This version of the User's Guide is current as of the date and version shown on the cover page. It is applicable only to the specified version of the code. Since the software development is continuing, and significant changes are constantly being made to the programs, the reader should be certain the guide is appropriate to the program version that will be used. Research is ongoing regarding the strengths and limitations of the YawDyn and AeroDyn codes. Users may wish to consult recent wind energy literature to improve their understanding of the code and its accuracy.

2.1 The YawDynVB Program

With version 10.0 of YawDyn we introduced a new Windows interface named YawDynVB. This "point-and-click" interface has been updated to version 2.0 for compatibility with YawDyn 11.0. YawDynVB can be used to create or modify input files, to run the YawDyn executable program, and to examine summary statistics or simple time-series plots of the results. The installation file can be downloaded from the NREL/NWTC Web site (http://www.nrel.gov/wind). A "readme" file and extensive help files are included in the file that you can download. There is no separate user's guide for YawDynVB. We expect that the help files, plus this User's Guide, will be sufficient for installing and running YawDynVB.

If you are running Windows NT or Windows 95 we recommend that you get a copy of YawDynVB to help you create your first YawDyn model, or to convert version 9.6 or 10.0 yawdyn.ipt files to version 11.0. You will still need much of the material that is presented in Section A of this Guide, but we hope that YawDynVB will be easier to use than the text-based method that is detailed herein.

3.0 Disk Files Included with YawDyn

Five files contain the source code for the YawDyn program. They are the main body of the program, YAWDYN.FOR, a file containing subroutines that are used by both YawDyn and ADAMS named AEROSUBS.FOR, a file containing full-field turbulence and output parameters named MODULES.FOR, and two include files AERODYN.INC and BEDOES.INC.

The primary data input file is called YAWDYN.IPT. If hub-height (steady or time varying) wind input is desired, an additional input file, referred to in this Guide as YAWDYN.WND, must be read by the program. This file contains hub-height wind data (details are provided below). An alternative method for simulating rotor operation in turbulence is also available. If turbulence is simulated, two additional data files must be present. This option is discussed further in a later section. Up to twenty airfoil data files are also used as inputs to YawDyn. Each file contains static and dynamic lift and drag information for up to twenty airfoils. (More than twenty airfoils and data files can be specified by changing MAXELEM in AERODYN.INC; see chapter 4.0 Memory Requirements). If desired, the files can also contain pitching moment coefficients. If they do, the blade aerodynamic pitching moment can be calculated. The multiple tables in one file can represent, for example, the same basic airfoil with different aileron or flap angle settings, or they can represent the same airfoil at different Reynolds numbers.

Up to three output files are created by YawDyn. The YAWDYN.OPT file is intended for printing a record of all the input conditions, and a summary of the basic model properties. File YAWDYN.PLT is tabular data intended for plotting the simulation results using a variety of commercially available graphics packages. An optional output file, ELEMENT.PLT, contains detailed aerodynamic data for each blade element. This file is primarily useful for interpreting results and debugging new simulations. It is generally too large for use in long simulations. As of version 11.0, each file created by YawDyn is tagged on the first line with the program name, version, and date, as well as the date and time of the file's creation.

To install YawDyn, copy all of the files from the distribution disk to your hard disk. You may want to create a new directory or folder to hold these files. First compile MODULES.FOR, which will create two dependency module files, FF_Wind.mod and Outputs.mod. Then compile and link the YAWDYN.FOR and AEROSUBS.FOR files. The two include (.INC) files (and module (.mod) files just created) must be in the same directory as the.FOR files while compiling. If you want to use the distribution copy of YawDyn without changes, you will need a compiler that will support large arrays (extended memory in DOS machines). See the Section 4.0 for ways to reduce the memory requirements so that the program will run with other compilers or computers with limited RAM.

4.0 Memory Requirements

The turbulence option in YawDyn increases its memory requirements substantially. In version 11.0, the turbulence array is dynamically allocated, so memory requirements are determined at run time if the full-field turbulence option is selected. If shortage of RAM makes it impossible to run YawDyn on your computer, try reducing the length of the simulation, which determines the size of the turbulence array.

Four parameters are assigned values in the AERODYN.INC file. MAXELEM can be changed by the user to increase the number of blade elements that can be analyzed. MAXCL can be edited to change the maximum length (number of angle-of-attack rows) of the airfoil tables. These parameters can also be reduced if RAM requirements are excessive. MAXBLD is the maximum number of blades on the rotor. It normally equals 3, but can be increased or decreased to meet a particular need. MAXTABLE controls the number of airfoil tables that can be used at a given blade station. If you are not using multiple tables (aileron simulations, for example), this value can be set to 1. All of these parameters are described in some detail in comment statements in the AERODYN.INC file.

5.0 Nomenclature and Sign Conventions

The YawDyn analysis is directed toward a wind turbine with the general configuration shown in Figures 5.1 and 5.2 or 5.3. The rotor can have 2 or more blades and the hub is rigid or teetering. The blade flap degree of freedom is modeled using an equivalent hinge and spring arrangement as shown in Figure 5.2 if the hub is rigid (not teetering). The teetering hub configuration is shown in Figure 5.3. When a teetering rotor is simulated, the blades are completely rigid. The only degrees of freedom are the teeter and yaw motion. Effects of undersling and the damping and stiffness characteristics of the teeter stop are included in the teetering model. Delta-three angle cannot be included in the model. The model assumes that all blades are identical in all respects *except* that each blade pitch angle is specified independently. (Blades can be specified to differ in their aerodynamic properties through the use of multiple airfoil tables, but this dependency on blade number must be programmed into the code by the user.)

The definitions of yaw angle (γ) and wind direction (δ) are shown in Figure 5.1. Both are measured in the same sense, with positive being clockwise when looking down. This is consistent with the compass directions generally used in reporting wind direction. Note however, that yaw is a rotation about the negative Z axis. The yaw angle is the angle the rotor makes with the ground (inertial) coordinate system, not with the instantaneous wind vector. Thus the yaw error (or difference between the compass rotor direction and wind direction) is $\gamma - \delta$. However, it is most common to use the program with the wind direction $\delta = 0$. Then the yaw angle and the yaw error are equal.

Tower shadow and vertical and horizontal wind shears can be simulated using YawDyn. Appendix A details the tower shadow model. Figure 5.4 shows a sketch of the wind shear conventions. A positive vertical wind shear causes an increase in wind speed with height above ground. A positive horizontal wind shear causes an increase in wind speed with increasing coordinate y.

The rotor can be downwind of the tower (positive L_S in Figure 5.1 or FORTRAN variable SL) or it can be upwind (negative L_S). The axis of rotation of the rotor can be tilted with respect to the horizontal (ground). The tilt angle τ is shown in its positive sense in Figure 5.1. Normally a downwind rotor will have positive tilt while an upwind rotor will have negative tilt (if the hub is raised above the level of the generator in either case).

The rotation of the rotor must be clockwise when viewed looking in the downwind direction. (This restriction does not apply in ADAMS.) If the rotor to be analyzed actually turns in the counterclockwise direction, the user must be careful interpreting the sign conventions. It is best to consider the position of the blade when it is advancing into the region of increased relative wind speed (due to yaw angle or wind shear) and adjust the signs of the yaw angle and wind shears to be appropriate to this condition.

An example may clarify this topic for YawDyn users. In the example, consider a downwind rotor that spins counterclockwise when viewed from a location that is upwind of the machine. In this case the rotor angular velocity vector (using the right hand rule) is directed from the hub toward the yaw axis and the rotation is opposite that used in the program. Consider also that the wind speed is higher on the left side of the rotor than on the right (when looking downwind). This situation is sketched in the views labeled "actual situation" in Figure 5.6. It is not possible to run the program with a negative (counterclockwise) rotor rpm, so other signs must be adjusted. With yaw and horizontal wind shear the blade will be advancing into the wind when the blade is vertical upwards (ψ=180°) and the yaw angle is *negative*. If the rotor spin were clockwise the advancing blade would be at ψ=180° when the yaw angle is *positive*, and the horizontal shear is *negative*. Thus the change in the sense of rotation requires a change in the sign of the yaw angle and the horizontal wind shear to achieve the same conditions for the blade. This is depicted in the views labeled "model equivalent" in Figure 5.6. To summarize, the actual situation in Figure 5.6 has counterclockwise rotor rotation, a negative yaw angle and positive wind shear. This is modeled with clockwise rotation, positive yaw, and negative horizontal shear. The goal at all times is to keep the orientation of the advancing blade correct.

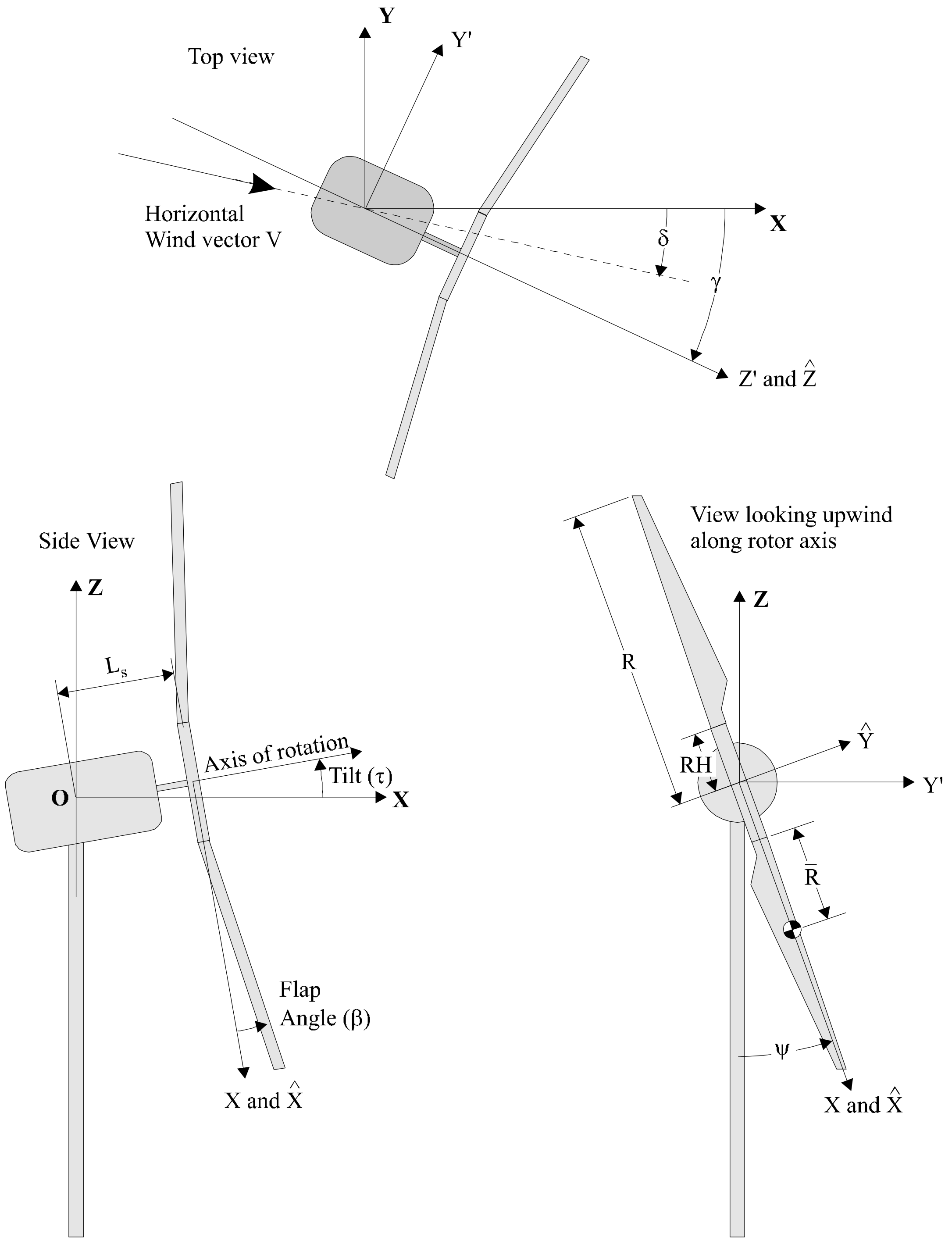


Figure 5.1 View of the HAWT defining selected terms and coordinate systems. All angles are shown in their positive sense. The bold **X,Y,Z** axes are fixed in space and are the coordinates in which the wind components are defined (VX, VY, VZ). Note that blade azimuth is zero when the blade is at the 6 o'clock position.

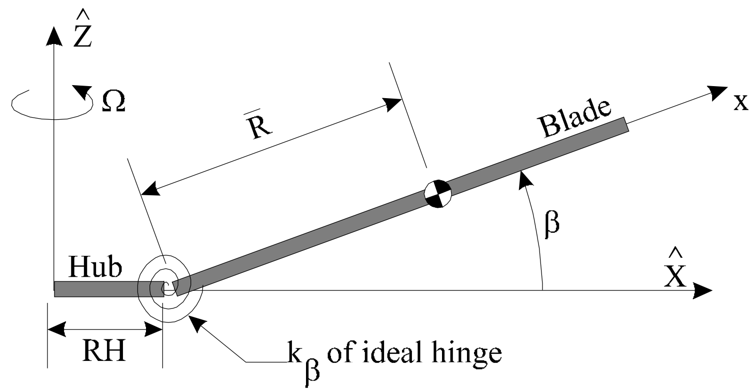


Figure 5.2 The equivalent hinge-spring model for the blade flap degree of freedom

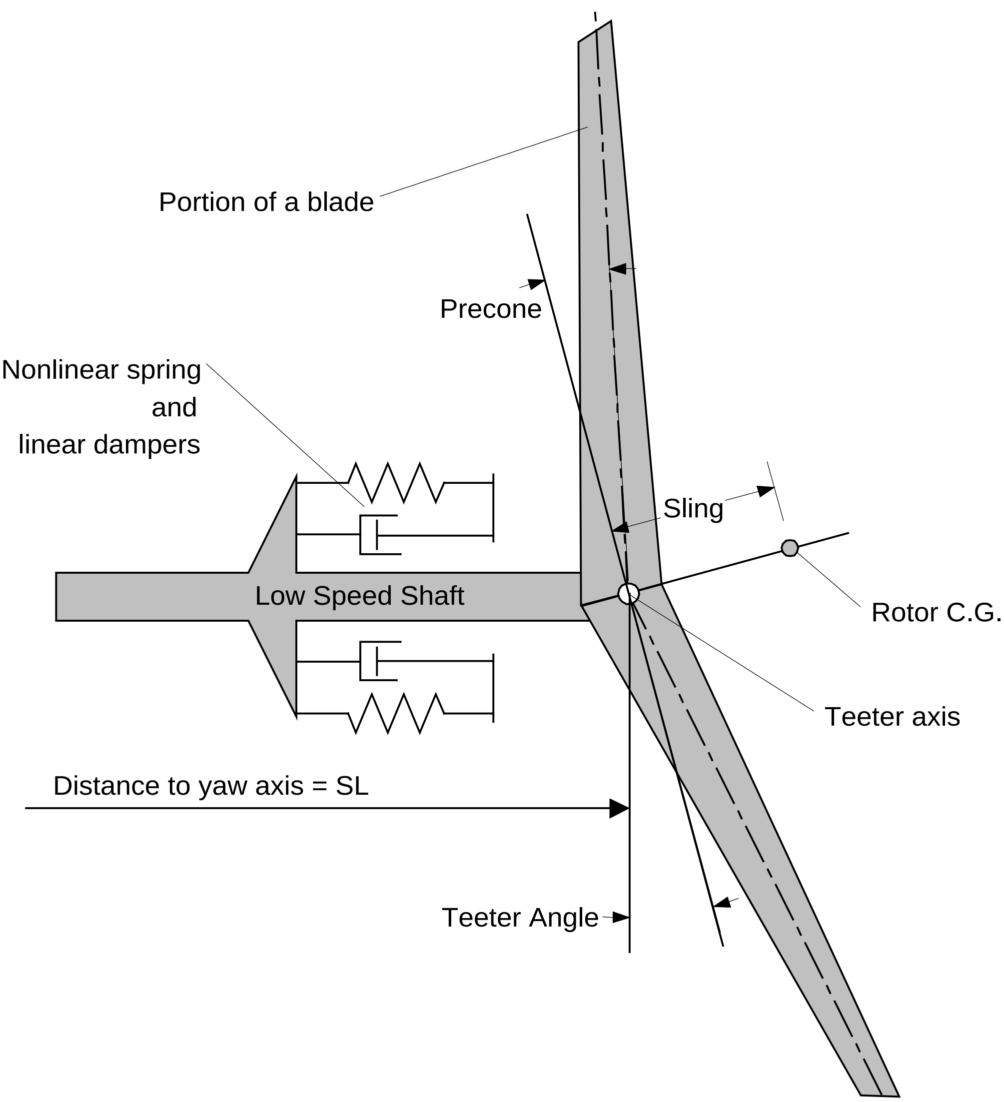


Figure 5.3 The configuration of the teetering hub model. The spring and damper are only active when the teeter deflection exceeds the angle TEE1. The flap angle of blade #1 is the sum of the preconing and the teeter angle.

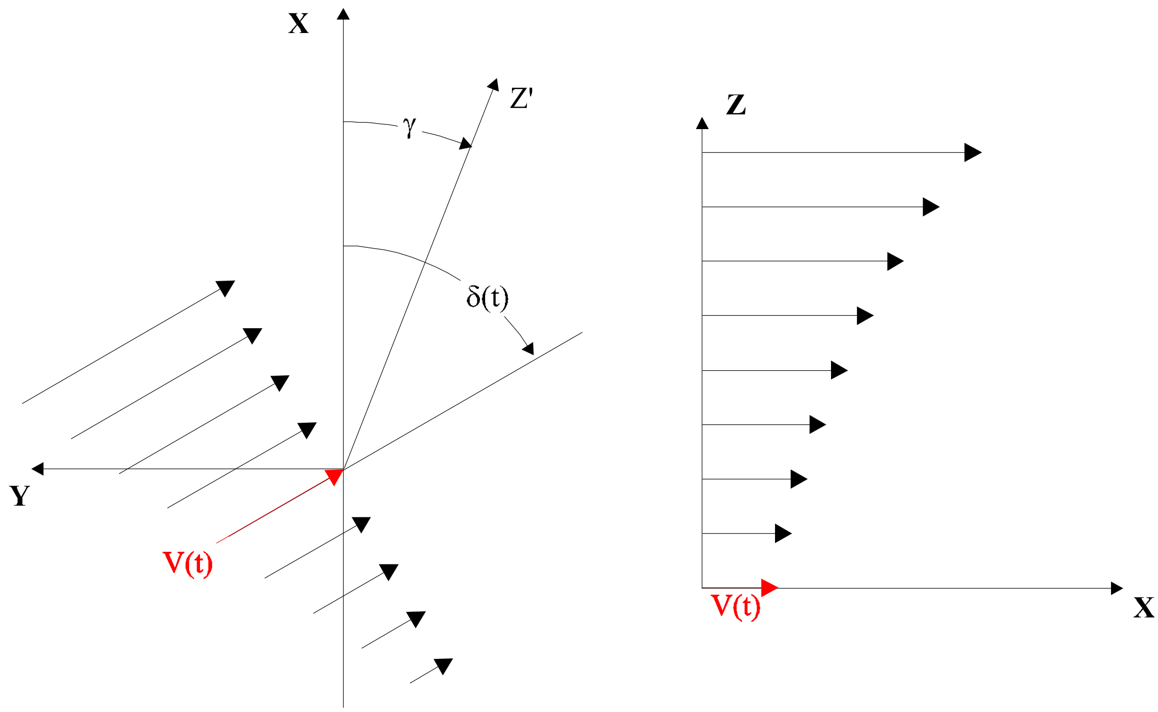


Figure 5.4 Wind shear models Horizontal shear in left sketch, Vertical shear in right sketch. Note the wind direction (δ) and yaw angle (γ) are both defined with respect to the **X** axis.

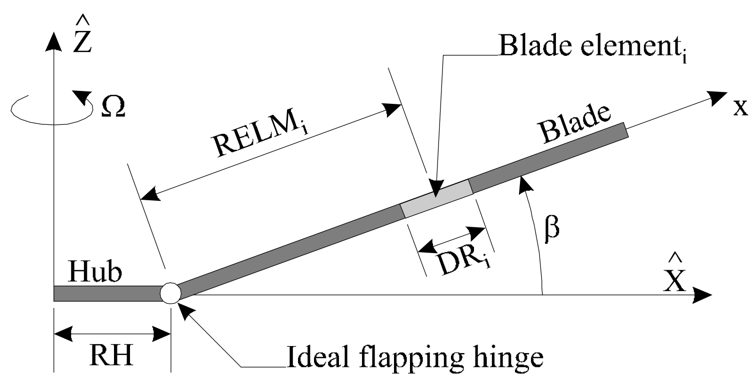


Figure 5.5 Sketch of the blade element geometry and nomenclature Note the element is identified by its length (DR_i) and its position ($RELM_i$) measured parallel to the blade span from the hinge axis.

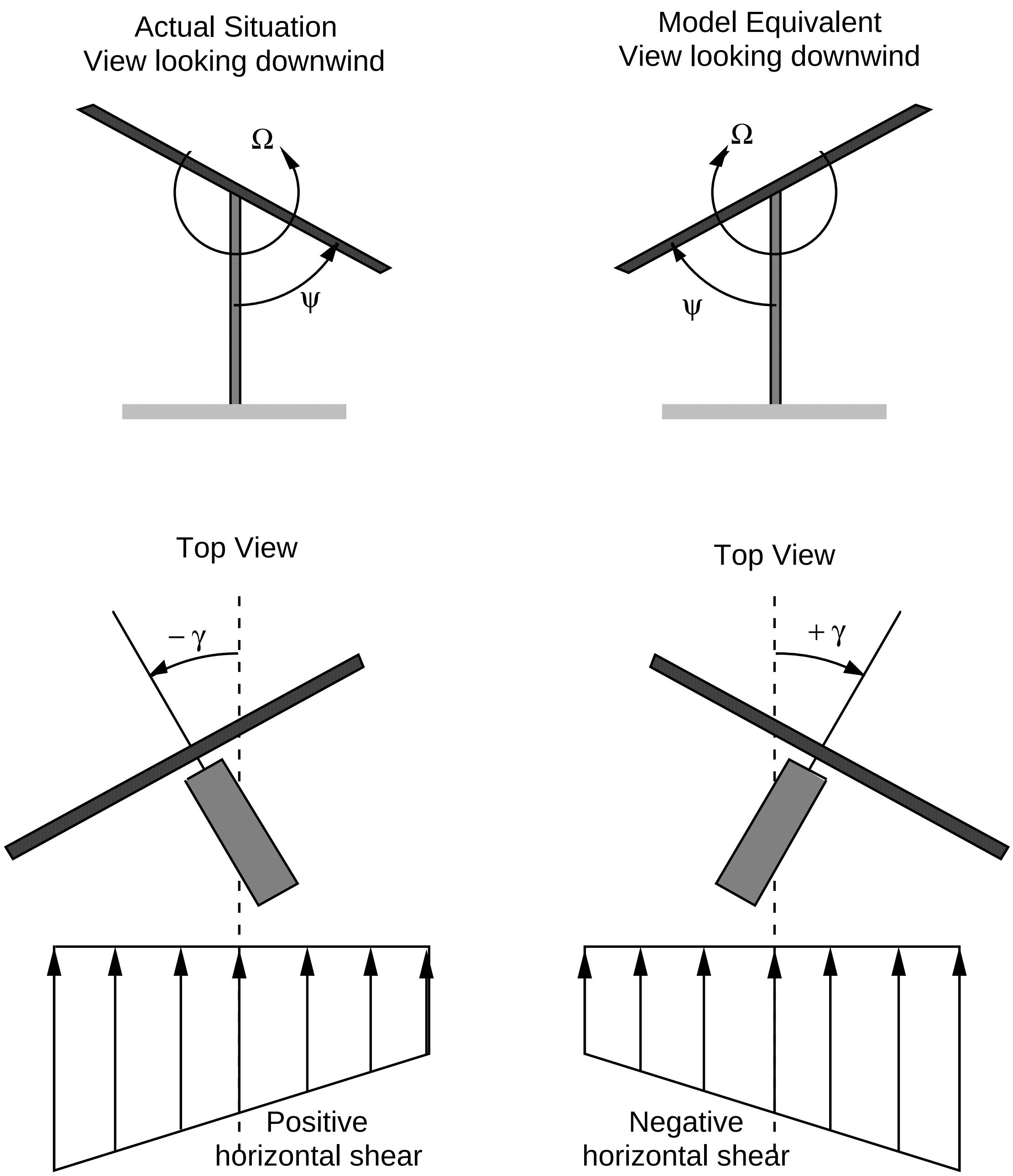


Figure 5.6 Views of example configuration with horizontal wind shear. Left half shows the actual configuration while the right side shows how that configuration can be modeled in YawDyn.

6.0 Input Data File Description

A sample YAWDYN.IPT input data file is given in Table 6.1. A text data file with this name and containing each of these items must be available in the directory or folder from which the program is run. The following paragraphs describe each of the input variables. The formatting is list-directed (or free). There are no restrictions on the spacing of the values other than the order of the variables on a line, the order of the lines, and the presence (absence) of a decimal point in a floating point (integer) value. Values on one line should be separated by one or more spaces or tabs. Each line, except the first and last, can be terminated with a text string to identify that line (lines that contain wind or airfoil file names should not contain other characters in the 80- to 100-character filename field). Each line must terminate with a return character. Each line must contain all of the variables specified for that line in the table below. Omission of a value that is not used by the program in that particular run may not result in a runtime error, but the line on which that parameter should be located must be present.

A line-by-line description of the input data file is given in Table 6.2. In this description, the engineering units for each parameter are listed for the program as they are used with English and SI (metric) units. It is quite simple to change the program units. Simply set the value of the unit parameter (line 10) to SI or ENGLISH in the input data file. Then the units input to the program must all be consistent with the unit system selected (e.g., kg, m, sec, N, deg or rad as listed below for SI).

The PRINT/NOPRINT flag found in the blade date table controls the contents of the element output data file. The volume of data makes it undesirable to print a record of all variables at all times for all blade stations during a simulation. Instead, the optional output file element.plt (aelement.plt for ADAMS) is created if desired. This file contains detailed blade element data such as angle of attack and aerodynamic coefficients and forces for selected elements. Also, the time steps in the simulation are generally shorter than needed for data output. The program will decimate all output by a factor of IPRINT (ID#31 in Table 6.2).

When YAWDYN.IPT is used with ADAMS rather than YawDyn, a number of the variables are not used in the calculations. However, to simplify use of the data with either program, the file format and contents are identical for both programs for the first half of the file. Those items that are not actually used in ADAMS follow the line that begins with the word END. These values need not be present if the data file is only to be used with ADAMS. Readers who are not using ADAMS can disregard any reference to ADAMS below.

Table 6.1 - Sample Input Data File for the NREL Combined Experiment Wind Turbine

```
Combined Experiment Baseline for YawDyn version 11.0
 INTERACT     Simulation mode:    INTERACTive, BATCH
 BEDDOES      Dynamic stall model:    BEDDOES, STEADY
 USE_CM       Aerodynamic pitching moment included?: USE_CM or NO_CM
 SWIRL        Induction Factor Model: NONE, WAKE, SWIRL
0.005         ATOLER, Tolerance for induction factor convergence (0.005)
 EQUIL        Dynamic inflow model:   DYNIN, EQUILibrium
 HH           Wind data file type: HH (Hub height) or FF (Full Field)
yawdyn.wnd
 ENGLISH      SIUNIT, select units: SI or ENGLISH
 0.1          Tower shadow deficit fraction
 3            Tower shadow width
 3            Number of blades
 0.002        Air density
 55           Hub height above ground
  4           Distance from yaw axis to hub
 0            Shaft tilt angle (deg)
 3            Rotor precone angle (deg)
 0.001        Aerodynamics time step (sec)
 1            Number of airfoil data files you wish to use
 S809_Cln.dat
 10           Number of blade elements per blade
  0.74      1.48       0      1.5      1    NOPRINT    RELM and Twist not used by ADAMS
  2.22      1.48       0      1.5      1    PRINT       but must be present in the dataset
  3.7       1.48       0      1.5      1    NOPRINT    RELM, DR,Twist, Chord, Airfoil#
  5.18      1.48       0      1.5      1    PRINT
  6.66      1.48       0      1.5      1    NOPRINT
  8.14      1.48       0      1.5      1    PRINT
  9.62      1.48       0      1.5      1    NOPRINT
  11.1      1.48       0      1.5      1    PRINT
  12.58     1.48       0      1.5      1    NOPRINT
  14.06     1.48       0      1.5      1    PRINT
END of ADAMS input, (the word END must appear in the first three cols)
 FIXED        Yaw Model: FREE or FIXED yaw system
 HINGE        Hub model:  HINGE, TEETER or RIGID
 SINGLE       Airfoil tables (or points between two tables) used: SINGLE or MULTIple
 0            TableID, Airfoil table used (see line 4 of airfoil file)
 1.667        Time duration of simulation (sec)
 200          Number of azimuth sectors used for integration
 5            Decimation factor for output printing
 0.01         TOLER, Trim solution tolerance (deg)
 72           RPM, rotor speed in revolutions per minute
 0            PsiInit, Initial rotor position (zero for Blade 1 down) (deg)
  14 14 14    Blade pitch angles
 0            Initial yaw angle (deg)
 0            Initial yaw rate (deg/s)
  3 3 3       Initial flap angles for each blade (deg)
  0 0 0       Initial flap rate for each blade (deg/s)
 1000         Mass moment of inertia about yaw axis
 3.34         Mass of one blade
 178          Mass moment of inertia of blade about hinge axis
 155000       Torsional stiffness of blade root spring
 5.44         RBAR, distance from hinge to blade c.g.
 1.7          RHinge, radius of rotor hub
 0            YawStiff, stiffness of yaw spring
 0            YawDamp, yaw damping coefficient
 0            YawFriction, constant friction moment at yaw axis
  0           Free teeter angle (deg)                 NOT USED
  0           Teeter stiffness, first or linear coeff.  NOT USED
  0           Teeter stiffness, coeff. of deflection^2  NOT USED
  0           Teeter damping coefficient              NOT USED
1,20,16,10,30,33,36,24,26,28
 The last line above is a list of output channels.  They are defined as follows:
 1 = Horiz. wind speed at hub, len/sec.
 2 = Horiz. wind direction at hub, deg.
 3 = Nacelle yaw angle, deg.

   portion deleted for brevity, see the sample file on the distribution disk

 45 = Blade 3 in-plane mom., kiloforce*len.
 46 = Blade 1 pitching mom., kiloforce*len.
 47 = Blade 2 pitching mom., kiloforce*len.
 48 = Blade 3 pitching mom., kiloforce*len.
```

Table 6.2 - Descriptions of YawDyn Input File Parameters

ID Number[1]	Units[2]	Description
1 TITLE	--	Any character string (up to 80 characters) to identify the system being analyzed. This also serves as an aid to identifying the contents of the data file.
2 BATCH or INTERACT	--	This value controls the simulation mode of the program. INTERACT will run the program as all past versions have run, waiting for user response when pause statements are encountered after errors or warnings. BATCH mode replaces pause statements with a time-delay function, allowing the simulation to continue after warnings and errors in the same manner as if the user had chosen to continue after a pause in INTERACT mode. This option was added in version 11.0 to prevent the program from "hanging" on pause statements during unattended operation. <u>These and subsequent 'string' values other than file names should be entered in **upper case only**</u>.
3 STEADY or BEDDOES	--	This value determines whether the Beddoes-Leishman dynamic stall model will be used. Enter BEDDOES for this dynamic stall model or STEADY for quasi-steady airfoil characteristics. We recommend using the BEDDOES model in most situations.
4 USE_CM or NO_CM	--	This value controls the option of calculating aerodynamic pitching moment. Enter USE_CM if you want to calculate the pitching moment. Enter NO_CM if you wish to ignore the pitching moment calculation. If you enter USE_CM, then you must provide pitching moment coefficients (CMs) in all of your airfoil data tables. If you enter NO_CM, then the CM values need not be present in your airfoil tables (but they can be present *provided you have only one airfoil data table in the file*).
5 WAKE, SWIRL or NONE	--	This value controls the wake or induced velocity calculation. There are three options, WAKE, SWIRL and NONE. This value should normally be SWIRL so that the axial and tangential induction will be analyzed. If WAKE is used, then only the axial induction will be calculated. If the value is NONE, then the induced velocity calculation will be completely bypassed and all induction factors will be zero. This option is available primarily to assist the debugging of new ADAMS models. We suggest that the first tests of a new ADAMS model ignore the wake to accelerate the calculations and eliminate the possibility of convergence problems in the induction factor iteration. A warning is printed to the screen when this value is NONE to remind the user this is a highly unusual situation.

[1] This column contains a sequential number, the ID number, and a name. The name represents the variable name in most cases. However, in the case of inputs that control a program option, the allowable inputs are listed.

[2] Units are specified for English system in the first line and SI units in the second line (if different). If SI units are desired the unit identifier (line 10) must be SI.

6 ATOLER	--	The tolerance used for convergence testing in the iterative solution to find the induction factor A. In earlier versions of the software, this value was always 0.005. This is a good default value that should be used unless there are compelling reasons to do otherwise. Some users may find it desirable to change this value to avoid convergence problems (with some loss of accuracy) or to speed the calculations. The value represents the maximum allowed difference between two successive estimates of A. That is, if the new estimate of A differs from the estimate from the previous iteration by an amount less than ATOLER, the solution has converged, and the last value of A is used. ATOLER is used for all induction factor calculations when using the EQUIL option (see next), but only for the trim solution in YawDyn when using the DYNIN option.
7 DYNIN or EQUIL	--	This input controls the dynamic inflow option. When the value is DYNIN, a modified Pitt and Peters dynamic inflow model is used to calculate the induction factor. This method replaces the blade element/momentum (BEM) method and time-lag method used prior to version 11.0. The direct calculation method of DYNIN is considerably faster than the iterative method of the EQUIL option. For more information on the dynamic inflow method, see Appendix E. A value of EQUIL assumes that the wake is always in equilibrium with the forces on a blade element. (The “quasi-steady” or equilibrium wake assumption.)
8 HH or FF	--	This input controls whether the YAWDYN.WND file or turbulence files will be the source of wind speed data. (Users of earlier versions of the code should note that the yawdyn.ipt file no longer contains wind speed or shear information.) If you wish to simulate steady wind conditions or simple Hub-Height, time-varying winds you should enter HH. If you wish to simulate Full-Field, 3-D turbulence you must enter FF. If you wish to allow the wind, pitch angles or other machine parameters to vary with time, then the desired values can be stored as a time series in file YAWDYN.WND (YAWDYN.WND is a generic name, the actual filename is entered on the next line). If you enter HH on this line, then the operating conditions are read from the data file whose name is given on the next input line. This makes it possible to run the program using actual values of operating conditions from test data or synthesized time series of wind conditions.

9 HHWindFile or FFWindFile	--	The actual filename for the file referred to in this guide as the YAWDYN.WND file or the prefix for the turbulence files. This name can be up to 100 characters long and can include the full pathname for the file. Any character in the first 100 columns of this line is considered a part of the file name. Leading and trailing spaces are ignored, but embedded spaces are not. If you selected the FF option in the previous line, then this line contains a character string that identifies the filenames where the turbulence data are found. The turbulence files must be in the *binary* format generated by the NREL SNLWIND-3D program. This string is the <u>leader</u> or prefix that is common to two file names. It can be up to 100 characters long and *any character in the first 100 columns of this line is considered a part of the file name.* The filenames must end in `.wnd` and `.sum`. These files contain the full-field wind data in binary format, and a summary of the simulated turbulence created by SNLWIND-3D, respectively. For example, consider the case of turbulence files named `9ms.wnd` and `9ms.sum`. These files might be in the 'myturb' directory on the C: drive. Then the characters `C:\myturb\9ms` would be entered on this line of data and could remind the user that the mean wind speed for these files is 9 m/s. Even though two data files will be read, only one line is used to enter the file name prefix. See documentation from NREL concerning the turbulence data files.
10 SI or ENGLISH	--	This input designates the system of units you are using for the input and output variables. If you enter SI then YawDyn works in the SI system (Newtons, kilograms, meters, seconds and their combinations). For example, moments of inertia will be kg-m^2. Outputs will be in N-m or kN-m, depending on your choice of outputs (see ID#54). If you enter ENGLISH then YawDyn works in the English system (Pound force, slugs, ft, seconds). In this case, moments of inertia will be in slug-ft^2. Outputs will be in ft-lbf or ft-klbf, depending on your choice of outputs (see ID#54). There is one exception to this rule: Power output of the rotor is always expressed in kW, regardless of the units system you select. All angles, such as pitch, preconing and angle of attack, are input and output in degrees regardless of the units system selected. ADAMS users must be certain that the units selected here are consistent with the units employed in their ADAMS data set. No units conversions are performed by the subroutines (excepting the conversion of forces and moments to kilo-forces and kilo-moments, if selected, for output).
11 TWRSHAD	--	A measure of the strength of the velocity deficit in the wake of the tower (tower shadow). The value is the magnitude of the fractional decrease in local wind speed at the center of the tower shadow. The deficit and the width (next line) are specified at the reference length SL (see below), which is essentially at the hub distance from the tower. Typical values are 0.0 to 0.2. The value should be zero for an upwind rotor.
12 SHADHWID	ft m	The half-width, b, of the tower shadow, measured at a distance SL from the tower. The tower wake width increases as the square root of the distance from the tower, and the wake strength decays inversely proportional to the root of the distance. See Appendix A for more details.
13 B	--	Number of blades, B=2 or more, except the teetering rotor must have B=2. The value of MAXBLD must be changed in the AERODYN.INC file for simulation of rotors with four or more blades.

14 RHO	slug/ft^3 kg/m^3	Ambient air density.
15 HH	ft m	Hub-height of the rotor above the ground. This is the height of the center of the hub, and is not equal to the tower height if TILT (ID#17) is nonzero.
16 SL	ft m	The distance from the yaw axis to the center of the hub. A positive value is used for a downwind rotor, a negative value for an upwind rotor. The teeter axis is always located at the center of the hub, though it need not coincide with the rotor center of gravity.
17 TILT	deg	The tiltangle of the rotor axis of rotation. The sign convention is <u>not</u> consistent with the coordinate system. That is, positive tilt is a rotation about the negative Y'-axis. A downwind rotor will have positive tilt in the normal situation where the hub is tilted upwards. An upwind rotor will normally have negative tilt (again, with the hub tilted upwards).
18 PC	deg	Blade precone angle. Coning is positive when the coning moves the blade tips downwind relative to the hub. (Positive coning normally gives centrifugal relief of blade root flap moments.)
19 DTAERO	sec	Time interval for aerodynamics calculations in ADAMS. This value is not used by YawDyn, but the line must be present in the data set. In an ADAMS simulation, the typical integration time step is quite small. The simulation can run faster and be more immune to numerical stability problems if a value for DTAERO is entered that is greater than the integration time step, but less than the time scale for the changes in aerodynamic forces. Typically, the aerodynamic forces should not be expected to change faster than the time it takes the blade to rotate 2-4°. For example, if a rotor runs at 30 rpm, it will take 0.02 sec for the blade to move 3.6°. If DTAERO is 0.02 sec, the aerodynamic calculations will repeat often enough to catch the true variations in loads, but several times slower than the integration time step. As with other time-step controls in numerical integrators, it is best to experiment with a range of values to determine the maximum value that will give the same results as the smaller values. Use of a very sharp tower shadow will normally require use of a small value for DTAERO. Since the integration time stepping is under the control of ADAMS, the aerodynamics calculations will be repeated after a time interval that is *at least* DTAERO (but less than DTAERO+ the integration time step). That is, the calculations are generally not repeated exactly at the interval DTAERO.
20 NUMFOIL	--	The number of different airfoil files that will be used to describe the blade elements. A maximum of 20 (or MAXELEM) files can be used. (See the section titled "Memory Requirements" for a discussion of MAXELEM.)

21 FOILNM	--	The name of the first data file (number 1) which contains the airfoil data. See the airfoil data file section below for a description of the contents of this file. *The filenames are limited to 80 characters and must be the only characters that appear in the first 80 columns of the line.* Trailing and leading blanks in the filename are ignored. If multiple airfoil files are used, successive lines will provide the filename for each of the other airfoil data tables in the same manner. Only one filename is entered on each line.
22 NELM	--	The number of blade elements per blade. A maximum of 20 (or MAXELEM) elements can be distributed along the blade.
23 RELM	ft m	This and the subsequent NELM-1 lines describe the details of each blade element for the aerodynamic analysis. The blade elements need not be equally spaced. They should be entered in order proceeding from inboard sections to outboard sections. The first entry in each line specifies the location of the center of the blade element. RELM is measured from the flapping or teeter hinge axis to the center of the element in the direction of the blade span. See Figure 5.5 for a sketch of the element geometry and nomenclature. This value is ignored when running ADAMS. The correct value is obtained from markers in the ADAMS data set.
23 DR	ft m	The second entry on each line is the length of the blade element, measured along the span of the blade. Use care selecting DR and RELM if you are not using equally spaced elements. The two values must be compatible, but YawDyn does not test fully for compatibility or contiguity of the elements.
23 TWIST	deg	The third entry on each line is the twist of the blade element. The twist is measured relative to the element for which the blade pitch is specified. The twist affects only one blade element while the pitch changes the angles of all blade elements, just as a full-span pitch control system would. The angle that a particular blade element's chord line makes with the plane of rotation is the sum of the pitch and twist angle for that element. The sign convention for TWIST is the same as for PITCH. The TWIST value is ignored when running ADAMS.
23 CHORD	ft m	The fourth entry on each line is the chord of the blade element. The planform area of the blade element equals CHORD*DR.
23 NFOIL		The fifth entry on each line is an integer between one and twenty (or MAXELEM) that determines which airfoil data file is to be used for each blade element. The first airfoil file listed in the YAWDYN.IPT is number 1, the next line specifies the name of the airfoil file number 2, etc. If a value of NFOIL=1, then the blade element will use data from the first airfoil file. If NFOIL=3, the airfoil data will be read from the third filename entered above, etc.

23 PRINT or NOPRINT	--	The sixth and final entry on each line is a string flag to control whether output is written for that element to the element.plt (aelement.plt for ADAMS) file. If the string PRINT is found, output is written for that element. If the string NOPRINT is found, or PRINT is not found, element data is not written. YawDyn responds to whichever string occurs first on each line. If neither string is found on a line, no data is written for that element. The element.plt file is not created if no element data is requested (i.e., PRINT is not found on any line).
24 END	--	This line is placed as a marker in the data set to identify the end of the input data that is required by ADAMS. The first three characters <u>must be END</u>. This line also provides a convenient means of checking that the data are self-consistent, e.g. that the number of lines of blade element data is correct, etc.
25 FIXED or FREE	--	An input to determine whether the simulation is for fixed yaw rate, or free-yaw operation. If you enter FIXED, the system operates at a fixed yaw rate equal to the initial yaw rate (Q(4), ID#37 below) specified further below in the file. For fixed yaw, use FIXED *and set the yaw rate to zero*. In the condition of a non-zero constant yaw rate, friction and damping are taken into account, but the yaw torsional stiffness is set to zero. If you enter FREE, then the system is constrained by the yaw torsional spring. If the spring stiffness (YAWSTF, ID#46 below) is zero, then the system is free-yawing.
26 HINGE, TEETER or RIGID	--	An input to determine whether the rotor has a rigid, flapping or teetering rotor. There are three possible entries, HINGE, TEETER, or RIGID. The RIGID option creates a model with only a yaw degree of freedom. This will allow a faster simulation for examining power output or other parameters that are not greatly influenced by blade flapping motion. A value of HINGE creates the flapping hinge model. If you enter TEETER then the rotor will have a teeter hinge at the center of the hub (a teetering rotor must have two blades).
27 SINGLE or MULTI	--	This value determines whether you will use a SINGLE airfoil table (or single interpolation point between two airfoil tables), or MULTIple airfoil tables (or multiple interpolation points between tables) during the simulation. If your airfoil files have only one table in them, this line is ignored. You should use SINGLE if you want to use a single table from a file containing multiple tables (or if you want to use a single interpolation point between two tables) for the duration of the simulation. <u>Use MULTI only if you supply your own algorithm</u> to move between airfoil tables during a simulation. Most YawDyn users will not need this option. It is useful for those who wish to implement control algorithms, or investigate roughness or Reynolds number effects in YawDyn.
28 TableID	(user defined)	This parameter selects the airfoil table (or the interpolation point between two tables) used for the simulation, provided there are multiple tables. If your airfoil files have only one table, this line is ignored. It corresponds directly to the TableID parameter on line 4 of the airfoil data file (see Table 9.2). Each airfoil file can contain up to 20 (or MAXTABLE) different airfoil tables. (Different tables may represent different aileron angles, or they can quantify the effect of some other variable such as Reynolds number or surface roughness.) The value of TableID determines which table is used from each file, or where the interpolation point between tables is, when the SINGLE option is selected above. In this case, enter a TableID parameter corresponding to the desired value on line 4 of your airfoil file. If you use MULTI above, this value may be used for the initial condition, if your algorithm requires one.

29 ENDTIME	sec	The total time which will be simulated in the solution. This value determines the number of rotor revolutions to be simulated according to the relation Total Time * RPM / 60=N. RPM is specified below (ID#33).
30 SECTOR	--	The rotor disc is divided into 'SECTOR' equally spaced, pie-wedge sectors for the time-integration. The value should be a positive, whole number. The time step is determined from the floating-point value SECTOR using the equation Δt = 60/(SECTOR*RPM). Typically 60-90 sectors are sufficient if the flap degree of freedom is neglected and 150-200 sectors are sufficient if the flap dof is included. As the blade stiffness increases in YawDyn the value of SECTOR must increase as well. If the program will not converge to a trim solution, increase SECTOR. When the program is run in free-yaw the value for SECTOR should be increased if a stiff blade is flapping in the simulation. A value between 600 and 1500 may be needed. The maximum value of SECTOR is determined by the 3rd dimension of the FETRIM array in the main program (file YAWDYN.FOR). In the current version of YawDyn the maximum value for SECTOR is 1500.
31 IPRINT	--	The decimation factor for data output. Typically, writing every 5th to 10th time step (IPRINT=5 or 10) will provide output data with adequate resolution.
32 TOLER	deg	The tolerance used in checking for a trim solution. The trim solution will be found when the root-mean-square difference in flap angles between two sequential rotor revolutions is less than TOLER for all blades. Typically, TOLER should be 0.01-0.02° for a rigid hub and 0.1-0.2° for a teetering hub. If the solution will not converge, try increasing SECTOR or, as a last resort, increasing TOLER.
33 RPM	rpm	Rotor rotation speed in revolutions per minute. Must be greater than zero; very small values should be avoided as they may cause problems as well.
34 PsiInit	deg	Initial azimuth position of the rotor at the start of the YawDyn simulation in the convention of YawDyn (see Figure 5.1). A value of 0 positions the rotor with blade 1 down (in the 6 o'clock position), which was the default starting position before this parameter was added in version 11.0. Since the rotor in YawDyn spins clockwise looking downwind, 90 degrees is blade 1 horizontal and left of the rotational axis looking downwind (the 9 o'clock position), etc.
35 PITCH(1)	deg	The pitch angles of each blade. Enter a value for each blade, separated by spaces or tabs. The sign convention follows the normal wind-turbine convention, i.e. positive pitch rotates the leading edge of the blade into the wind (toward feather). Negative pitch tends to move the blade toward stall. The pitch value rotates the entire blade. The values need not be the same for all blades, and a value must be entered for each blade.
36 Q(3)	deg	Initial yaw angle for the solution. When a yaw drive stiffness is specified and the program is run for "free yaw", this angle also specifies the undeflected position of the torsion spring. For "fixed yaw" (FIXED (ID#25) with zero yaw rate specified below) this is the angle at which the nacelle is positioned for the entire simulation.

37 Q(4)	deg/s	For "free yaw," this is the initial yaw rate for the solution. For "fixed yaw," this parameter must be set to zero. If you specify FIXED (ID#25), and Q(4) is non-zero, the simulation will run at a *constant yaw rate*, with the yaw torsional spring stiffness set to zero (friction and damping are taken into account).
38 QP array	deg	Initial flap angle for each blade. Enter a value for each blade, separated by spaces or tabs. (For a teetering rotor, only the value for blade #1 is read.) The initial flap and flap rate (next line) are important to the efficient convergence to a rotor trim solution. It is suggested that when a rotor is analyzed for the first time, the flap angles should all equal the precone angle and the flap rates should all be zero. This will usually result in a slow but accurate convergence to a trim solution. When the trim solution is found, the values of flap and flap rate are output to the CRT. These values can be used in subsequent runs of the program to significantly reduce the time required to find the trim solution. When the rotor is RIGID (ID#26), these values are ignored by the program.
39 QP array	deg/s	Initial flap rate for each blade. Enter a value for each blade, separated by spaces or tabs. (For a teetering rotor, only the value for blade #1 is read.) When the rotor is RIGID (ID#26), these values are ignored by the program.
40 YI	slug-ft^2 kg-m^2	Mass moment of inertia about the yaw axis of the main frame, nacelle and those portions of the rotor that are not participating in the flap or teeter motion (e.g. the hub of a rigid rotor). YI represents the total moment of inertia of all the yawing mass except the blades. For a teetering rotor YI does not include the hub inertia because the hub is moving with the blades as they teeter. For a rigid or flapping rotor YI includes the hub mass. When the model is FIXED (ID#25), this value is ignored (no yaw accelerations are calculated for fixed yaw rate operation).
41 BM	slugs kg	The mass of one blade. When the rotor has a teetering hub the mass of one blade plus half of the hub mass should be entered in this location. That is, for a teetering rotor only, this value represents one-half of the rotor mass.
42 BLINER	slug-ft^2 kg-m^2	The blade mass moment of inertia about the flap axis. For a flapping hinge or rigid rotor, this value is the inertia about the hinge axis. If the rotor is teetering then BLINER represents one-half the moment of inertia of the entire rotor (including hub and any concentrated masses) about the teeter axis (see Figure 5.3).
43 FS	ft-lbf/rad N-m/rad	The torsional spring constant of the equivalent flapping hinge spring at the blade root. This value is named k_β in the reports and literature and in Figure 5.2. When using the TEETER (ID#26) option this line is ignored. When using the RIGID (ID#26) option, this value is only used to calculate the value of the flap natural frequency.
44 RB	ft m	The distance along the blade from the hinge axis to the blade center of gravity. For a teetering rotor, this is the distance from the teeter axis (see Figure 5.3) to the center of gravity of the combination of one blade and one-half of the hub.
45 RH	ft m	This value affects both the dynamic and aerodynamic analysis. It is the distance from the axis of rotation to the hinge axis, measured perpendicular to the axis of rotation (RH is the hub offset). When TEETER (ID#26) is selected, this value is ignored and RH is set to zero for the teetering rotor.

46 YAWSTF	ft-lbf/rad N-m/rad	The torsional spring constant of the yaw drive or yaw brake system. This variable can be used to represent an equivalent stiffness of the yaw drive system and the tower (the overall effective torsional stiffness between the nacelle and ground). The value is only used when "free yaw" is simulated (ID#25 = FREE), but the line must always be present in the data file. If the actual system stiffness is very high, then the system should be run in "fixed yaw" (ID#25 = FIXED *and* zero yaw rate (i.e., Q(4) (ID#37) = 0.0)).
47 AV	ft-lbf-sec N-m-sec	The linear yaw damping coefficient. The yaw moment (ft-lbf) due to mechanical damping on the yaw axis is AV multiplied by the yaw rate in radians/sec. This value is used in both "free yaw" and "fixed (non-zero) yaw rate" simulations. Although not used for "fixed yaw," this line must always be present.
48 AF	ft-lbf N-m	The sliding friction yaw moment. A constant yaw moment due to friction. The moment always opposes yaw motion. Note this is not a friction coefficient. This value is used in both "free yaw" and "fixed (non-zero) yaw rate" simulations. Although not used for "fixed yaw," this line must always be present.
49 TEE1	deg	The teeter angle at which the first contact with the teeter "stop" is made. No mechanical teeter moment is applied at the hub if the absolute value of the teeter angle is less than TEE1. For teeter angles greater than TEE1, a nonlinear spring and a linear damper are active. See Figure 5.3 for a sketch of the teetering hub configuration. Used for a teetering rotor only, but this line must be present in the file.
50 SPRNG1	ft-lbf/rad N-m/rad	The first (linear) coefficient in the quadratic equation that describes the teeter spring or "stop". The moment applied by the teeter spring is given as $M=SPRNG1*\delta+SPRNG2*\delta^2$ Where δ is the spring deflection in radians and the sign is chosen as appropriate for the direction of deflection. Used for a teetering rotor only, but this line must be present in the file.
51 SPRNG2	ft-lbf/rad^2 N-m/rad^2	The second coefficient in the equation that describes the teeter spring. SPRNG1 and SPRNG2 determine the shape of the parabolic spring which represents the teeter stop. Used for a teetering rotor only, but this line must be present in the file.
52 TDAMP	ft-lbf-sec N-m-sec	The coefficient of the linear teeter damping. The damper is active for all teeter angles greater than TEE1. The teeter moment (ft-lbf) due to mechanical damping on the teeter axis is TDAMP multiplied by the teeter rate in radians/sec. Used for a teetering rotor only, but this line must be present in the file.
54 Output List	--	A numeric list of the channels to be output to the yawdyn.plt file. The channels available are listed at the end of the sample input file provided, as well as Table 6.3. This is the last line read by YawDyn. You must list the channels you want to output. At least one channel must be listed in order for YawDyn to run. Commas, spaces, tabs, or other delimiters should be used to separate items in the list. The line must not contain any comments.

Table 6.3 - List of Available Output Channels

Channel Number	Description	Units - SI (English)
1	Horizontal wind speed at hub center	m/sec (ft/sec)
2	Horizontal wind direction at hub center	deg
3	Nacelle yaw angle	deg
4	Nacelle yaw rate	deg/sec
5	Blade azimuth angle (0 when blade 1 down)	deg
6	Blade azimuth angle (0 when blade 1 up)	deg
7	Teeter angle	deg
8	Teeter rate	deg/sec
9	Aileron angle	deg
10	Blade 1 flap angle	deg
11	Blade 1 flap rate	deg/sec
12	Blade 2 flap angle	deg
13	Blade 2 flap rate	deg/sec
14	Blade 3 flap angle	deg
15	Blade 3 flap rate	deg/sec
16	Rotor power	kW
20	Nacelle yaw moment	N-m (ft-lbf)
21	Nacelle yaw moment	kN-m (kft-lbf)
22	Hub moment	N-m (ft-lbf)
23	Hub moment	kN-m (kft-lbf)
24	Rotor thrust	N (lbf)
25	Rotor thrust	kN (klbf)
26	Lateral hub force	N (lbf)
27	Lateral hub force	kN (klbf)
28	Vertical hub force	N (lbf)
29	Vertical hub force	kN (klbf)
30	Out-of-plane bending moment for blade 1	N-m (ft-lbf)
31	Out-of-plane bending moment for blade 2	N-m (ft-lbf)
32	Out-of-plane bending moment for blade 3	N-m (ft-lbf)
33	In-plane bending moment for blade 1	N-m (ft-lbf)
34	In-plane bending moment for blade 2	N-m (ft-lbf)
35	In-plane bending moment for blade 3	N-m (ft-lbf)
36	Pitching moment for blade 1	N-m (ft-lbf)
37	Pitching moment for blade 2	N-m (ft-lbf)
38	Pitching moment for blade 3	N-m (ft-lbf)
40	Out-of-plane bending moment for blade 1	kN-m (kft-lbf)
41	Out-of-plane bending moment for blade 2	kN-m (kft-lbf)
42	Out-of-plane bending moment for blade 3	kN-m (kft-lbf)
43	In-plane bending moment for blade 1	kN-m (kft-lbf)
44	In-plane bending moment for blade 2	kN-m (kft-lbf)
45	In-plane bending moment for blade 3	kN-m (kft-lbf)
46	Pitching moment for blade 1	kN-m (kft-lbf)
47	Pitching moment for blade 2	kN-m (kft-lbf)
48	Pitching moment for blade 3	kN-m (kft-lbf)

7.0 The YAWDYN.WND Data File

If the HH option in the YAWDYN.IPT file is selected, the program will look for a tabulated time-series of operating conditions in a file with the generic name YAWDYN.WND. In this guide the file is referred to using the name YAWDYN.WND. The actual name of the YAWDYN.WND file is entered in the YAWDYN.IPT file and can be any path/filename up to 100 characters long. (If the FF option is selected, then the YAWDYN.WND file need not be present.) In the current version of the program the values of wind speed at the hub (V), wind direction (DELTA), vertical component of wind (VZ), horizontal wind shear (HSHR), power law vertical wind shear (VSHR), linear vertical wind shear (VLinShr) and gust velocity (VG) are entered in tabular form as a function of time (TDAT). A description of each column in the YAWDYN.WND file is provided in Table 7.1.

If you wish to simulate turbine operation in constant winds you may enter only one line of data in the YAWDYN.WND file.

The first lines of the file can be "comment" lines. A comment line must appear before any data in the file and it must contain the ! character (generally, though not necessarily in the first column). Any number of comment lines can be used at the beginning of the file, but no comments can be embedded in the wind data lines.

By making relatively simple changes to the source code, users can modify the list of parameters that will be read from the YAWDYN.WND file. FORTRAN READ statements which access this file are found in three locations in the program. The first two are in subroutine YAWIN, and the third is in subroutine GETWND. All statements are of the form:

```
READ(91,*) TDATD(2), VD(2), DELTAD(2), VZD(2), HSHRD(2), VSHRD(2), VLinShrD(1), VGD(2)
```

If desired, the list of variables can be shortened or extended to meet particular requirements. All that is required is to change these READ statements and any lines in the GETWND subroutine that alter values read from the file. Linear interpolation is used to obtain wind values at times that are between entries in the wind file. If changes are made to the parameters that are used in the distribution copy of YawDyn, the user must take care to change all lines of the GETWND subroutine pertaining to the interpolation.

One parameter that might be added is blade pitch. If pitch is added, one additional change is required. Since the program retains pitch angles for each blade independently, the YAWDYN.WND file must contain pitch data for each blade [i.e. `READ(91,*) TDATD(2), VD(2), ..., (PITCH(I),I=1,NB)`].

It can be seen from the READ statement that either spaces or tabs can be used to separate the tabular values in YAWDYN.WND. In the YAWDYN.WND file, all values for a particular time (TDATD) must be on one line, and each line must end with a return character. The time step between lines need not be constant. Linear interpolation is used for simulated time that is between two values of TDATD in the wind data file. If a simulation runs longer than the time length of the wind file, the simulation continues using the last line of the wind file as a steady wind condition for the remainder of the simulation.

A utility program named IECWind is included in the YawDyn distribution disk. This program generates YAWDYN.WND files for the discrete gust conditions specified by the 2nd edition IEC wind turbine design standard. (Note that some of the IEC extreme wind conditions have changed from the 1st to the 2nd edition standard.) Be sure to use IECWind for YawDyn 11.0 (IECWind 2.0 is for use with earlier versions of YawDyn). There is also a Windows version of the IECWind code called WindMaker for YawDyn 11.0 (WindMaker 1.2 is for earlier YawDyn versions), which is available form the NREL web site. IECWind or WindMaker can be quite helpful when analyzing the variety of gust conditions that must be considered during turbine design. A sample input data file for IECWind is also included on the disk (no input file is used with WindMaker). Both IECWind and WindMaker are distributed with a `readme.txt` file. This file, plus a copy of the IEC standard, provides the documentation needed to use the program.

Table 7.1 – YAWDYN.WND File Column Descriptions

Column	Parameter	Units	Description
1	TDAT	sec	Time at which the conditions on the current line are specified to occur. The first value should be zero, with subsequent values increasing monotonically. Intervals between time values need not be constant. Wind conditions between specified TDAT values are linearly interpolated. If the simulation time duration exceeds the last value of TDAT, the final value of each parameter is held constant for the remainder of the simulation.
2	V	m/s ft/s	This hub-height wind speed represents the total horizontal wind component. Units must be consistent with the selection made in yawdyn.ipt on line 10.
3	DELTA	deg	The wind direction of the horizontal component specified above, with zero aligned with the zero yaw angle (see Figure 5.4).
4	VZ	m/s ft/s	The vertical wind speed component is specified with the convention positive up. This value is assumed uniform over the rotor disc (i.e. it is unaffected by any specified shear values).
5	HSHR	--	The horizontal wind shear parameter represents the linear variation of wind speed across the rotor disc. Typical values are $-1. < \text{HSHR} < +1.$ and represent the wind speed at the blade tip on one side of the rotor, minus the wind speed at the blade tip on the opposite side of the rotor, divided by the hub-height wind speed (V). The shear is in the direction perpendicular to the hub-height wind vector specified by DELTA above. See Figure 5.6 for sign convention.
6	VSHR	--	The vertical power law shear is the exponent of a power-law shear profile. It is used to determine the wind speed, V_z, at any height, z, based on the hub-height, z_{hub}, and hub-height wind speed, V_{hub}, using the equation: $$V_z = V_{hub} (z/z_{hub})^{VSHR}$$ A typical value is 0.14 representing a $1/7^{th}$ power-law profile. Normally you should use either VSHR or VLinSHR (see next), not both (set the parameter you do not wish to use to zero).
7	VLinSHR	--	The linear vertical shear parameter works in the same way as HSHR but in the vertical plane across the rotor disc. It represents the wind speed at the blade tip at the top of the rotor, minus the wind speed at the blade tip at the bottom of the rotor, divided by the hub-height wind speed (V). Normally you should use either VLinSHR or VSHR (above), not both (set the parameter you do not wish to use to zero).
8	VG	m/s ft/s	The gust velocity is parameter used to add a constant horizontal wind speed component across the entire rotor disc. <u>This parameter is not influenced by any shear values.</u> This parameter is seldom used and can be set to zero for most cases. Changes in wind speed are normally specified using the hub-height wind speed parameter, V.

8.0 The Turbulence Data Files

The YawDyn subroutines can use simulated or actual wind data that represent all three components of the wind vector varying in space and time. This permits a detailed simulation of the rotor moving through a wind field with the appropriate scales and correlation of atmospheric turbulence. Two files, one binary wind data file and one summary file, must be in the specific form generated by the NREL program SNLWIND-3D. This program is also available from the NREL web site.

The components of the wind vector are expressed in the inertial coordinate system that has its origin on the yaw axis, at the hub height of the rotor. See Figure 5.1 to see the location of this **XYZ** coordinate system. A grid of fixed points (much like a vertical plane array of anemometers) is located in the **YZ** plane and centered at the hub height. The three velocity components are available at each grid point as a function of time. The subroutines interpolate in all three spatial dimensions (using a convection velocity to get a time shift for the **X** dimension) to obtain the wind velocity vector at each blade element at each time step.

The program reads and stores the turbulence files into memory at the start of execution. The total number of samples that can be stored, hence the time duration that can be simulated, is determined by the length of the simulation. See section 4.0 for information on memory requirements for YawDyn if you experience trouble running long turbulence simulations.

In earlier versions of the code a bicubic interpolation routine slowed the simulations considerably. In version 9.1 and later the interpolation is linear, yielding simulations with turbulence nearly as fast as those with steady wind and giving the same results as the much slower bicubic interpolation method.

9.0 The Airfoil Data Files

Up to twenty (or MAXELEM, if the parameter was changed by the user) different airfoil sections can be specified along the blade span. Each airfoil file can contain up to 100 (or MAXCL) angle-of-attack entries for each of up to 20 (or MAXTABLE) tables. A sample data file is shown in Table 9.1. A line by line description of this file is presented in Table 9.2. Each different airfoil section is described in a separate data file as identified in the YAWDYN.IPT file. The following table describes the format of the input data. All data are free-format. Comments can be included at the end of any line to serve as a reminder of the contents of that line. Any line that is for use by the dynamic stall model must be present even if dynamic stall is not considered in the simulation.

Appendix C describes a utility program called FoilCheck that may help with the preparation of airfoil data files.

Table 9.1 - Sample Airfoil Data File for the NREL Combined Experiment Wind Turbine

```
S809 Airfoil, OSU data at Re=.75 Million, Clean roughness
NREL/TP-442-7817 Appendix B, Viterna used aspect ratio=11
   1           Number of airfoil tables in this file
     .00       Table ID parameter
   15.30       Stall angle (deg)
     .00       No longer used, enter zero
     .00       No longer used, enter zero
     .00       No longer used, enter zero
    -.38       Zero lift angle of attack (deg)
   7.12499     Cn slope for zero lift (dimensionless)
    1.9408     Cn at stall value for positive angle of attack
    -.8000     Cn at stall value for negative angle of attack
    2.0000     Angle of attack for minimum CD (deg)
     .0116     Minimum CD value
 -180.00   .000  .1748  .0000
 -170.00   .230  .2116  .4000
 -160.00   .460  .3172  .1018
 -150.00   .494  .4784  .1333
 -140.00   .510  .6743  .1727
 -130.00   .486  .8799  .2132
 -120.00   .415 1.0684  .2498

   portion deleted for brevity, see the sample file on the distribution disk

  -50.00  -.486  .8799  .2132
  -40.00  -.510  .6743  .1727
  -30.00  -.494  .4784  .1333
  -20.10  -.560  .3027  .0612
  -18.10  -.670  .3069  .0904
  -16.10  -.790  .1928  .0293
  -14.20  -.840  .0898 -.0090
  -12.20  -.700  .0553 -.0045
  -10.10  -.630  .0390 -.0044
   -8.20  -.560  .0233 -.0051
   -6.10  -.640  .0131  .0018
   -4.10  -.420  .0134 -.0216
   -2.10  -.210  .0119 -.0282
     .10   .050  .0122 -.0346
    2.00   .300  .0116 -.0405
    4.10   .540  .0144 -.0455
    6.20   .790  .0146 -.0507
    8.10   .900  .0162 -.0404
   10.20   .930  .0274 -.0321
   11.30   .920  .0303 -.0281
   12.10   .950  .0369 -.0284
   13.20   .990  .0509 -.0322
   14.20  1.010  .0648 -.0361
   15.30  1.020  .0776 -.0363
   16.30  1.000  .0917 -.0393
   17.10   .940  .0994 -.0398
   18.10   .850  .2306 -.0983
   19.10   .700  .3142 -.1242
   20.10   .660  .3186 -.1155
   30.00   .705  .4784 -.2459
   40.00   .729  .6743 -.2813
   50.00   .694  .8799 -.3134
   60.00   .593 1.0684 -.3388
   70.00   .432 1.2148 -.3557
   80.00   .227 1.2989 -.3630
   90.00   .000 1.3080 -.3604
  100.00  -.159 1.2989 -.3600
  110.00  -.302 1.2148 -.3446
  120.00  -.415 1.0684 -.3166
  130.00  -.486  .8799 -.2800
  140.00  -.510  .6743 -.2394
  150.00  -.494  .4784 -.2001
  160.00  -.460  .3172 -.1685
  170.00  -.230  .2116 -.5000
  180.00   .000  .1748  .0000
```

Table 9.2 - Descriptions of Airfoil Data File Parameters

Line	Position	Name	Units	Description
1	1	TITLE(1)	--	Up to 40 characters of text to identify this data file. This title will be written to the screen when YawDyn executes to remind the operator which airfoil tables are being used.
2	1	TITLE(2)	--	Up to 40 characters of any text to identify this data file.
3	1	NPHI	--	The number of different airfoil tables contained in this file. If airfoils with ailerons are in use, this value represents the number of aileron angle settings for which aerodynamic coefficient data are provided.
4	1...NPHI	AILRN	--	The parameter that identifies each airfoil table (the table ID). Examples include aileron angle (from which the variable name was derived), Reynolds number, etc. Linear interpolation is done between tables based upon the value of the table ID that is desired.
5	1...NPHI	ALPHAS	deg	The next few lines of input pertain to the dynamic and static stall characteristics of the airfoil for use in the dynamic stall models. They must be present, though the values will be ignored, when the dynamic stall option is not selected. The first parameter is the static-stall angle-of-attack of the airfoil. We have found that this value is very important to simulation accuracy. Engineering judgment must be exercised in the selection of the stall angle. Airfoils with “flat-topped” stall characteristics should generally use a value near the mid range of the flat region. One value must be listed for each of the airfoil tables in this file.
6	0.0		--	Reserved for future use and backward compatibility. YawDyn does not use this value, but the line must be present in the file. Enter a value of 0.0.
7	0.0		--	Reserved for future use and backward compatibility. YawDyn does not use this value, but the line must be present in the file. Enter a value of 0.0.
8	0.0		--	Reserved for future use and backward compatibility. YawDyn does not use this value, but the line must be present in the file. Enter a value of 0.0.
9	1...NPHI	ALPHAL	deg	The zero-lift angle-of-attack of the airfoil. This parameter is used only in the Beddoes dynamic stall model.
10	1...NPHI	CNA	--	The static C_N (approximately equal to C_L) curve <u>slope</u> near zero lift. This dimensionless value is critical to the success of the Beddoes model and <u>must</u> be consistent with the tabular data that follow. We recommend using a least-squares fit to the linear portion of the C_N data to determine this value. An “eyeball” fit is not accurate enough in most situations. This value is used only by the Beddoes dynamic stall model. This value is dimensionless (ΔC_N/radian).

11	1...NPHI	CNS	--	The value of C_N at positive static stall. This is the nominal stall for positive and increasing angles-of-attack. This value is typically 1.0-3.0 and occurs at angle-of-attack near 15° to 30° (the higher values may be observed inboard on a rotating blade). This value is used only by the Beddoes dynamic stall model. We have found better correlation with test results when we use the value of CN extrapolated from the linear portion of the CN curve to the stall angle ALPHAS. This gives better results than using the actual CN value at the stall angle.
12	1...NPHI	CNSL	--	The value of C_N at the negative static stall angle of attack. This is "stall" for negative and decreasing angles-of-attack. This value is typically -1.0 and occurs at angles-of-attack near -10° or -20°. This value is used only by the Beddoes dynamic stall model.
13	1...NPHI	AOD	deg	The angle of attack for the minimum drag coefficient (C_{Dmin}). This value is used only by the Beddoes dynamic stall model.
14	1...NPHI	CDO	--	The minimum drag coefficient of the airfoil. This value is used only by the Beddoes dynamic stall model.
15+	1	AL	deg	The angle of attack for the first point in the lift & drag coefficient table. The table must be written in order of increasing angle of attack. It must cover the entire range of angles of attack that might be encountered by any blade element. It is preferable to use a table for angles between -180° and +180°. If YawDyn attempts to lookup a value outside the table range, program execution will stop with an error message. Note that inboard blade elements can easily encounter angles of attack approaching ±180° if the rotor is operating at a large yaw angle. Care should also be taken to ensure that the values of the coefficients are the same at +180° and -180° to avoid a discontinuity. There is a limit of 100 entries in each airfoil table. (This can be increased by changing the MAXCL PARAMETER in the AERODYN.INC include file.) YawDyn obtains all airfoil data by linear interpolation from the tables provided. The user must be certain that adequate resolution is available in the table to make linear interpolation accurate. The points need <u>not</u> be equally spaced, so it is advisable to enter many points near stall and fewer points at very large (positive or negative) angles.
15+	2, 4, 6, .. or 2, 5, 8, ..	CL	--	The static lift coefficient corresponding to the angle of attack entered on this line. Many lines such as this are entered to completely specify the lift coefficient vs. angle-of-attack curve.

15+	3, 5, 7, .. or 3, 6, 9 …	CD	--	The drag coefficient corresponding to the angle of attack entered on this line. Note the CL, CD and CM (if used) must be specified for the same angles of attack.
15+	4, 7, 10, .. or not present	CM	--	The pitching moment coefficient corresponding to the angle of attack entered on this line. Note the CL, CD and CM (if used) must be specified for the same angles of attack. This value must be present if the USE_CM option is selected. It can, but need not, be present and will be ignored if the NO_CM option is selected for a single-table airfoil file. *If you are using multiple tables in your airfoil file, then you must not have CM values in the table if you have selected the NO_CM option.*

10.0 User Operation at Runtime and the YAWDYN.PLT file

No user input is required while the program is running unless warnings or errors are generated. When running in INTERACTIVE mode, the program will wait for a user response after errors and warnings before continuing. When running in BATCH mode, the program will post an error and warning message, then continue after 30 seconds. The CRT will display information on the status of the calculations and a few statements about the run conditions so that the calculations can be stopped if the desired conditions are not being run. The lines below are typical of what will be seen as the program executes (though the details may not match the sample input file provided above). The `Courier` font is used for information that will be sent to the CRT. Annotations are shown in the Helvetica font.

The search for the trim solution will continue until the "RMS ERROR" values for all blades are less than the "TOLER" value from the input file. If 50 trim revolutions are run before the solution converges to a trim condition the calculation will be aborted. If this occurs, use different initial conditions or try a larger tolerance for the trim criteria.

First the program displays the TITLE information. It then proceeds with the calculations and writes information to the screen concerning progress and results of those calculations. (Some details have changed slightly since this CRT transcript was copied into the User's Guide, so your display will not match this one perfectly. The general content and the meaning of the various outputs have not changed.)

```
Running YawDyn (v11.0, 03-Mar-1998).

Combined Experiment Baseline for YawDyn version 11.0

 Simulation running in INTERACTIVE mode
 FIXED-YAW ANALYSIS
 Dynamic inflow theory not used in the analysis


 *******************************************
  Only 1 line present in windfile.
  Simulation will use steady wind conditions.
  Hub height wind speed =    30.00000
 *******************************************

 Only 1 line in wind file, steady wind conditions used

   BLADE ROTATING NATURAL FREQUENCY (P#) =   4.060922

 RUNNING          400 POINTS
 WITH    200.0000     POINTS PER REVOLUTION
```

```
 TOTAL TIME DURATION SIMULATED (SEC) =    1.666667

 Seeking trim solution for flap DOF
                      AZMTH=  40.  FLAP=  2.8← Status of the trim solution
                      AZMTH=  85.  FLAP=  3.1       ← for Blade 1
                      AZMTH= 130.  FLAP=  2.9
                      AZMTH= 175.  FLAP=  3.0
                      AZMTH= 220.  FLAP=  2.9
                      AZMTH= 265.  FLAP=  3.0
                      AZMTH= 310.  FLAP=  2.9
                      AZMTH= 355.  FLAP=  2.9
                      AZMTH=  45.  FLAP=  2.9
                      AZMTH=  90.  FLAP=  3.0
                      AZMTH= 135.  FLAP=  3.0
                      AZMTH= 180.  FLAP=  3.0
                      AZMTH= 225.  FLAP=  2.9
                      AZMTH= 270.  FLAP=  2.9
                      AZMTH= 315.  FLAP=  2.9
                      AZMTH=   0.  FLAP=  2.9
 TRIM REVOLUTION  2
   BLADE #1  FLAP=   2.91     FLAP RATE=   -1.32    RMS ERROR= 0.04940
   BLADE #2  FLAP=   2.95     FLAP RATE=    0.82    RMS ERROR= 0.03632
   BLADE #3  FLAP=   2.97     FLAP RATE=   -0.41    RMS ERROR= 0.01612
                      AZMTH=  45.  FLAP=  2.9
```

…etc the output is shortened for brevity…

```
 TRIM REVOLUTION  4
   BLADE #1  FLAP=   2.87     FLAP RATE=   -1.03    RMS ERROR= 0.00408
   BLADE #2  FLAP=   2.95     FLAP RATE=    1.13    RMS ERROR= 0.00415
   BLADE #3  FLAP=   2.96     FLAP RATE=   -0.33    RMS ERROR= 0.00101
```

…trim criteria have been satisfied, rms errors for all blades is below TOLER = 0.01…

```
 Initial values for transient solution:
 BLADE     FLAP   FLAP RATE
   1     2.88    -1.03
   2     2.95     1.13
   3     2.96    -0.33
```

…now proceed with the transient solution…

```
 Starting transient solution...
 T=    0.8  AZ=   0.   YAW=    0.0   YR=   0.0   FLAP=  2.9  FR= -1.0
 T=    1.7  AZ=   0.   YAW=    0.0   YR=   0.0   FLAP=  2.9  FR= -1.0


Simulation Timing:

 Total Clock Time:      130.620 seconds
 Startup Clock Time:    126.940 seconds
 Transient Clock Time:    3.680 seconds
 Transient Sim Time:      1.667 seconds

 Sim/Clock Time Ratio:    0.453

Finished.
```

The output file YAWDYN.PLT is useful for plotting predictions as a function of time. The first line of this file declares that YawDyn (including version number and date) created it, and the date and time it was created. A line of column headings follows this, which in turn is followed by the time series data. Tabs are used to separate the columns, allowing many spreadsheet and graphics software packages to easily read the file.

The YAWDYN.PLT file can contain a variety of response data, dependent upon the channels selected for output in the yawdyn.ipt file. The column headings may not be self-explanatory, so check the outputs selected in yawdyn.ipt for clarification.

The thrust, and lateral and vertical hub forces are net aerodynamic forces. They are defined with respect to the rotor plane. Thrust is normal to the plane, the horizontal and vertical force components are in the plane of the rotor. These forces do not include inertial forces, and are therefore not the net thrust or lateral forces. The in-plane blade bending moments include only aerodynamic and gravity forces. Other inertial forces are neglected even though they might be significant in some situations. The out-of-plane moments include all aerodynamic and inertial loads. The moments are defined with respect to the plane of rotation and will only equal the "flap" moment or "edge" moment for zero pitch.

The optional ELEMENT.PLT file contains wind and aerodynamic data. This is also a tab-delimited file with a program declaration line at the top, followed by column headings and then data. All three components of wind speed are output in two forms. The first form is the three velocity components measured at the hub location (the wind that would be there in the absence of the turbine). The second set of three columns are the wind speeds at the location of the outermost element of blade number 1 as it moves through the wind field. Thus the wind field is the "rotationally sampled" wind. The wind speed is the ambient wind after it is modified by wind shears and/or tower shadow (if applicable). Effects of the rotor induced velocity are not reflected in the output wind speeds. Usually the x-component of wind speed will show once-per-revolution variations due to wind shears in addition to a higher frequency, negative pulse caused by the tower shadow. The y- and z-components will normally be zero unless a wind direction or vertical wind is specified, or a turbulence simulation is run. Since the z-component of the wind is not altered by wind shear or tower shadow, it will always equal the rotationally-sampled input value. Aerodynamic data are output for those blade elements requested by the user. The section angle of attack, lift and drag coefficients, normal and tangential aerodynamic force, and induction factor are produced to provide details about the aerodynamic behavior of the blade.

YawDyn also produces a summary file named YAWDYN.OPT, which reiterates the inputs, and provides other useful information. A Sample YAWDYN.OPT file is shown in Table 10.1.

Table 10.1 - Sample YAWDYN.OPT file from Program YawDyn 11.0
Using input file given in Table 6.1.

```
This file was generated by YawDyn (v11.0, 03-Mar-1998) on 24-Mar-1998 at 14:40.
 Combined Experiment Baseline for YawDyn version 11.0

   Simulation run in INTERACTIVE mode
   ENGLISH UNITS USED FOR INPUT AND OUTPUT
   INITIAL WIND SPEED AT HUB          =   30.00000
   INITIAL WIND DIRECTION DELTA (DEG) =  0.0000000E+00
   VERTICAL COMPONENT OF WIND SPEED   =  -1.000000

   PITCHING MOMENTS WERE CALCULATED

   AIR DENSITY                   =   2.0000001E-03
   ACCELERATION DUE TO GRAVITY   =    32.20000

   LINEAR HORIZONTAL WIND SHEAR
     INITIAL SHEAR COEFFICIENT   =   0.1000000
   POWER LAW VERTICAL WIND SHEAR
     INITIAL POWER LAW EXPONENT  =   0.1400000
   LINEAR VERTICAL WIND SHEAR
     INITIAL LINEAR SHEAR COEFF. =   0.0000000E+00

   TOWER SHADOW CENTERLINE VELOCITY DEFICIT  =   0.1000000
   TOWER SHADOW HALF-WIDTH AT REFERENCE POS. =    3.000000


   ROTOR RADIUS  =   16.47972
   HUB RADIUS    =   1.700000
   HUB HEIGHT    =   55.00000
   INITIAL PITCH ANGLES (DEG) =   14.00000       14.00000       14.00000
   BLADE CENTER OF GRAVITY    =   5.440000
   YAW AXIS-TO-HUB DISTANCE   =   4.000000
   NUMBER OF BLADES           =   3.000000
   PRE-CONING ANGLE (DEG)     =   3.000000
   ROTOR TILT ANGLE (DEG)     =  0.0000000E+00

   INITIAL BLADE 1 AZIMUTH POSITION =  0.0000000E+00
   ROTOR SPEED (RPM)                =   72.00000

   MASS OF BLADE                    =   3.340000
   BLADE FLAP MOMENT OF INERTIA  =     178.0
   NACELLE MOMENT OF INERTIA     =       0.0

   YAW STIFFNESS COEF.  =   0.0000000E+00
   YAW AXIS FRICTION    =   0.0000000E+00
   YAW AXIS DAMPING     =   0.0000000E+00

   BLADE STIFFNESS COEFF.                =   155000.0
   BLADE NATURAL FREQUENCY (HZ)          =   4.696518
   Blade rotating natural frequency (P#) =   4.060922

   Wind and other operating parameters read from
      the following hub-height wind file:
     yawdyn.wnd

   Winds not read from full-field wind files

   INITIAL FLAP ANGLE (BLADE 1) (DEG)  =    3.000000
   INITIAL FLAP RATE (BLADE 1) (DEG/S) =   0.0000000E+00
   INITIAL YAW ANGLE (DEG)  =   0.0000000E+00
   INITIAL YAW RATE (DEG/S) =   0.0000000E+00

   FIXED YAW OPERATION
   FLAP DEGREE OF FREEDOM WAS CONSIDERED
   ANALYSIS OF A RIGID-HUB ROTOR

   TOLERANCE FOR TRIM SOLUTION
      CONVERGENCE TEST =  9.9999998E-03

   WAKE (INDUCTION FACTOR) WAS CALCULATED IN THIS SIMULATION
   CONVERGENCE TOLERANCE FOR INDUCTION FACTOR =  4.9999999E-03
```

```
ANGULAR INDUCTION FACTOR WAS CALCULATED IN THIS SIMULATION
DYNAMIC INFLOW WAS NOT CONSIDERED IN INDUCTION FACTOR CALCULATION

PRINT INTERVAL (TO PLOT FILE) =           5

DYNAMIC STALL MODEL WAS USED IN CALCULATIONS

BEDDOES DYNAMIC STALL PARAMETERS:

CN SLOPE          7.1250
STALL CN (UPPER)  1.9408
STALL CN (LOWER) -0.8000
ZERO LIFT AOA    -0.3800
MIN DRAG AOA      2.0000
MIN DRAG COEFF    0.0116

  VORTEX TRANSIT TIME FROM LE TO TE    11.00000
  PRESSURE TIME CONSTANT               1.700000
  VORTEX TIME CONSTANT                 6.000000
  F-PARAMETER TIME CONSTANT            3.000000


AIRFOIL TABLES USED IN THE ANALYSIS
ID   FILENAME
 1   S809_Cln.dat


ELEM     R       DR   TWIST ANGLE  CHORD  AIRFOIL
                         (DEG)             ID

  1    0.740    1.480    0.000     1.5000     1
  2    2.220    1.480    0.000     1.5000     1
  3    3.700    1.480    0.000     1.5000     1
  4    5.180    1.480    0.000     1.5000     1
  5    6.660    1.480    0.000     1.5000     1
  6    8.140    1.480    0.000     1.5000     1
  7    9.620    1.480    0.000     1.5000     1
  8   11.100    1.480    0.000     1.5000     1
  9   12.580    1.480    0.000     1.5000     1
 10   14.060    1.480    0.000     1.5000     1

 FLAP MOMENT IS THE BLADE DEFLECTION TIMES THE SPRING STIFFNESS
 YAW MOMENT IS NET APPLIED MOMENT


 FILE element.plt CONTAINS BLADE ELEMENT DATA
```

SECTION B: User's Guide to AeroDyn for ADAMS

11.0 Introduction

This section of the Guide describes the use of the AeroDyn FORTRAN subroutines for the aerodynamic analysis of horizontal-axis wind turbine rotors. The subroutines were sponsored by the National Renewable Energy Laboratory and written at the University of Utah for use with two codes that perform wind turbine dynamics analysis. The two codes are: 1) ADAMS®, a commercially available software package for Automatic Dynamic Analysis of Mechanical Systems written by Mechanical Dynamics, Inc. (MDI, Ann Arbor, MI); and 2) YawDyn, a code written at the University of Utah for analysis of wind turbine yaw dynamics (in the public domain). This Section pertains to the aerodynamics model for the ADAMS code. Section A of this Guide describes the use of the YawDyn code and creation of the input data files for YawDyn. Readers should understand the material from Section A before reading Section B.

This guide is written for the engineer/analyst who is familiar with wind turbine dynamics and aerodynamics as well as the ADAMS basics. Rudimentary knowledge of the application of user-written subroutines in ADAMS is also required. Information on ADAMS can be obtained from MDI's documentation or courses. Background information on the YawDyn code and the aerodynamics analysis employed in YawDyn can be found in a National Renewable Energy Laboratory (NREL) final report (Hansen, 1992).

12.0 Requirements

To compile, link and run ADAMS with the aerodynamic subroutines the following items are required (these requirements pertain to the Windows NT operating system. Compilers and version numbers will vary for other platforms):

Purchased items:

1) ADAMS® version 9.0 and all hardware required to run ADAMS (Mechanical Dynamics, Inc., Ann Arbor, MI)

2) A FORTRAN 90 compiler (DIGITAL Visual Fortran, version 5.0a).

Public domain items:

3) Seven files containing the source code (GFOSUB.FOR, AEROSUBS.FOR, MODULES.FOR, REQSUB.FOR, SENSUB.FOR, AERODYN.INC, and BEDOES.INC). These files are available from the University of Utah as a part of the normal distribution of the software package.

4) This User's Guide. This guide is available from the author at the University of Utah.

User-supplied items:

5) Several data files that are created by the user. Up to five different types of input data files are required to use the ADAMS aerodynamics routines. 1) The ADAMS data set that creates the structural model. 2) A file named YAWDYN.IPT that contains basic data such as blade aerodynamic characteristics and analysis control parameters. This file is always required. 3) File(s) that contain airfoil lift and drag data in addition to dynamic stall data. This data is also always required. 4) An optional file called YAWDYN.WND that contains time-varying parameters such as wind speed or blade pitch. 5) Another set of optional files that contain wind turbulence data. The content and format of all of these files are the primary subjects of this User's Guide.

The ADAMS data set is not discussed in detail in this User's Guide. That information is available in the extensive MDI manuals. However, the aerodynamics routines place some requirements on the content of the data set. Three types of statements are required in the data set:

a) GFORCE statements that apply the aerodynamic forces.

b) A variety of MARKERS.

c) A SENSOR statement.

These requirements are detailed in the sections that follow.

13.0 Background

ADAMS simulates the dynamics of a structure which is described using an "ADAMS data set". The turbine model can be quite complex and will analyze system dynamics such as coupled blade and tower vibrations. To model a wind turbine, ADAMS must obtain information about the aerodynamic forces on the blades. This is accomplished with the AeroDyn "user-written" subroutines that compile and link with ADAMS. These subroutines make it possible to model the aeroelastic interactions between the blade motions and the aerodynamic forces.

The YawDyn program was developed for analysis of the blade and yaw loads and motions of a HAWT. It contains a complete subroutine package to model rotor aerodynamics. The package was developed at the University of Utah under an earlier contract with NREL and is in the public domain. This aerodynamics package was the starting point in the development of the aerodynamics subroutines for ADAMS. The development and validation of the ADAMS wind energy package started in 1991 and is continuing. Thus the methods and software are constantly evolving as new information becomes available. It is important that users of the software provide feedback and suggestions to NREL and the University of Utah to expedite the improvements to the code.

The evolutionary nature of this software and the need for constant comparison with other codes has led to a decision by the authors to maintain complete compatibility between the ADAMS and the YawDyn versions of the subroutines. In the first release of the software, this had both positive and negative implications. Subsequent revisions of the code have sought to eliminate most of the negative aspects of this compatibility. Now most of the parameters that are required only for YawDyn are placed at the end of the input data file. Anyone desiring to use ADAMS alone can prepare just the first portion of the data file. Anyone wishing to use both ADAMS and YawDyn must prepare the complete file. One advantage of the compatibility is that the subroutines are identical for both programs. This makes it faster and less expensive to maintain and update the code. Also, the aerodynamic input files for the two codes are identical. This facilitates comparisons of the codes and makes it easy for experienced YawDyn users to learn the ADAMS input file format. Finally, the use of identical subroutines will offer researchers the ability to explore the importance of various structural modes of vibration with identical aerodynamics analyses.

AeroDyn uses one of two aerodynamics models, depending on the value of the EQUIL/DYNIN flag yawdyn.ipt. The EQUIL option activates a modified blade-element/momentum (BEM) theory model. It includes standard BEM theory with multiple airfoil tables and arbitrary size, location and twist of up to 20 elements per blade. Corrections are applied to the induced velocity field when the rotor is skewed with respect to the wind (using a modified dynamic inflow theory of Pitt and Peters). The DYNIN option activates dynamic inflow theory equations based on a modified Pitt and Peters model. Skewed wake conditions are handled inherently by the dynamic inflow model. The dynamic stall model of Beddoes and Leishman is available as an option in the code. Hansen (1992) provides details of the dynamic stall analysis method in the YawDyn final report. More recent references are listed at the end of this Guide to provide sources of more detailed information on the code and methods.

Aeroelastic feedback is easily modeled using ADAMS. The absolute velocity vector of any blade element is available as a function of time from ADAMS (using INFFNC or INFARY subroutines). Obviously, the velocity can be the result of rotor rotation, structural vibration, or other rotor motions from causes such as yaw rate or tower bending. When the blade velocity is (vector) subtracted from the wind velocity at the same location, all the relative velocity components required for aerodynamic analysis are available. The aerodynamic forces thereby become functions of the blade vibratory motions and the feedback loop between forces and motions is closed.

Wind inputs to the ADAMS model are the same as those used in YawDyn simulations. For details on the wind input options, see chapter 7.0 - The YAWDYN.WND Data File, and chapter 8.0 - The Turbulence Data Files, of this Guide.

13.1 A Suggested Strategy for Modeling Wind Turbine Systems

ADAMS with AeroDyn is an extremely versatile and powerful tool. As with all powerful tools, there are many ways the analyst can approach the task of developing a complex model. We are convinced that the model development will proceed faster, with greater confidence, and more useful final results if the following general strategy is employed.

Step 1. Create a simple YawDyn model of the turbine. This may seem counterproductive if your ultimate goal is analysis of structural degrees of freedom that are not available in YawDyn. But it provides the opportunity to develop a basic understanding of the system, particularly the aerodynamic behavior. Little effort is wasted because the YawDyn input files are also needed for the ADAMS model. YawDyn should be used until you are confident the mean loads and power output are correct. This usually requires “tweaking” the airfoil data files and checking all of the input variables. Doing this with YawDyn offers the advantages of simplicity and speed. *This is a very simple and useful step that is often ignored until the complete ADAMS model is running but giving mysterious results.*

Step 2. Evaluate the need for a more complex model. You may be surprised and discover that YawDyn is adequate for your needs. If this is the case you can save a great deal of time by using the simple model. Often this is not the case and you will need to develop an ADAMS model.

Step 3. If this is your first ADAMS model, we suggest creating a turbine in ADAMS that is identical to the YawDyn model you have just been using. That is, create a rotor with a flapping hinge and rigid blade, a rigid tower, and fixed yaw. ADAMS WT® makes it quite easy to create such a model. Run this model to be certain you get the same results as you see in YawDyn. We have done this a number of times and consistently find the two programs agree within a few percent when the models are identical. This will help you gain confidence that you have correctly modeled the turbine in ADAMS and that you understand the marker and data input requirements. You will also gain experience running and debugging a complete ADAMS data set without the complexity of a complete system model. The power train can be modeled with a MOTION statement to specify a constant rotor rpm. Simulations will run quite fast and the results will be relatively easy to interpret.

Step 4. Modify the model from Step 3 to use flexible blades. This will allow you to simulate all blade degrees of freedom without the additional difficulties involved in modeling power train and tower dynamics. You can “tune” the structural model to match natural frequencies known from finite element analysis or modal testing of the blade. Your model now includes many blade motions that YawDyn cannot analyze. You have arrived at this point by a series of incremental improvements and have a relatively small area in which to search if your model requires debugging. You will probably need to model the power train using a torque-speed equation to avoid large accelerations during startup. You may need to change the integrator parameters in ADAMS to achieve accurate results.

Step 5. Update your model to include all degrees of freedom you feel may be important. Again, changes can be incremental to make debugging relatively easy. It is very important to remember that a good model will reflect and sharpen the analyst’s understanding of the system. This is much more important than using all of the degrees of freedom that are possible, simply because the software gives you that capability.

14.0 Adams Markers

ADAMS obtains information about the blade position and velocity by using an INFARY or INFFNC subroutine in conjunction with markers. Some markers are required by the aerodynamics routines and they must have specific marker numbers, locations and orientations. We have tried to minimize the number of "hard-wired" marker numbers in the code, but some are essential. This section describes the markers that are required in the ADAMS data set. If the user wishes, it is possible to change the marker numbers in the subroutines to match them with the ADAMS data set. However, this requires a thorough understanding of the subroutines and is not recommended.

We have been using a marker and part numbering scheme that has helped us navigate through our own models and, more important, makes it easier for others to understand a model. This numbering system is not required by the subroutines, but we do find it useful and include it in this guide for your information. Table 14.1 lists the number ranges and their associated turbine components.

The required markers are shown in Figure 14.1. They are discussed in the sections that follow. If you are using ADAMS WT to create your model, these markers should be created for you. Additional markers are used in the sample REQSUB.FOR file that is distributed with the software. These are not required for any operation other than generating program output and are not included in Figure 14.1 or detailed in this guide.

14.1 Ground marker, ID = 1

The ground marker is attached to the ground part and must have ID number 1. It is oriented with z vertical upwards and x in the nominal downwind direction. It must be located at ground level at the origin of the GROUND local part reference frame. (The wind need not actually be in the x direction, but all wind components are specified in this ground marker coordinate system.)

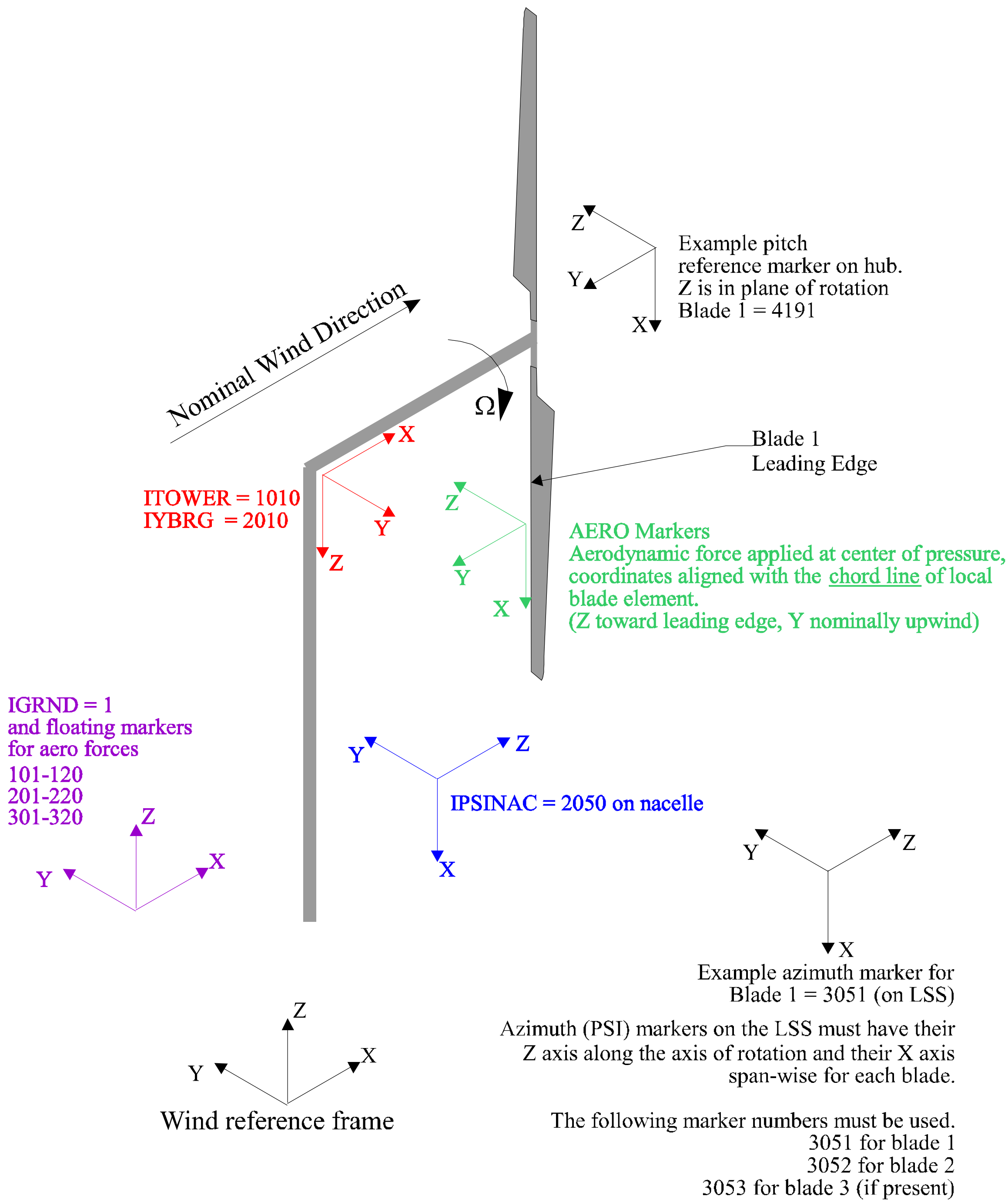


Figure 14.1 Required marker identification and orientation

Table 14.1 - Marker and Part Number Ranges (Suggested)

Part and Marker Numbers	Components
1 - 999	Ground, other markers fixed to ground
1,000 - 1,999	Tower and guy cables
2,000 - 2,999	Nacelle, yaw bearing, mainframe
3,000 - 3,999	Power train: Generator, gearbox, shafts, etc.
4,000 - 4,999	Hub, teeter bearing, teeter stops, etc.
11,000 - 11,999	Blade #1
21,000 - 21,999	Blade #2
31,000 - 31,999	Blade #3

14.2 Aerodynamic force (AERO) markers, IDs selected by user

These markers identify the point of application of the GFORCE aerodynamic forces. One marker is required for each GFORCE (or blade element, in the terminology of the rotor aerodynamicist). Each marker must be aligned with the chord line of the local element airfoil (see below) with the Z-axis pointed toward the leading edge of the blade element, as shown in Figure 14.2. Thus all AERO markers will be aligned with the chord line of the blade element at the location of the marker. The pitch angle of each blade element is obtained by ADAMS by requesting the angle between the AERO marker and the pitch reference marker on the hub (see below). The PITCH + TWIST values familiar to users of YawDyn or PROP are both reflected in the location of the AERO markers. The PITCH and TWIST values from the YAWDYN.IPT file are ignored by ADAMS. This change was implemented in version 8.1 and is quite different from earlier versions. The Y-axis is oriented nominally upwind such that the Y component of the aerodynamic force is usually negative. The X-axis will be directed either inward or outward along the span of the blade, depending upon the direction of rotor rotation. If the blade rotates clockwise (counterclockwise) when viewed looking downwind, the X-axis will be pointed outward (inward) along the blade span.

Twist of the structural elements of the blade is defined using BEAM or other similar markers in the ADAMS data set. This structural twist must not be confused with the aerodynamic twist mentioned above.

Up to 20 blade elements can be defined for each blade with a maximum of three blades (these limits can be changed, see Section 4.0). Each element must have an aerodynamic force marker, a floating marker on the ground (see section 14.3 below), and a GFORCE statement that references the two marker ID’s. For example, the following statements might appear in the data set to identify one GFORCE. The MARKER/101 definition given in section 14.3 must also be in the data set.

```
MARKER/11010, PART=10000, QP=0.74, 0.0, 0.0, REU = 0.0D, 90.0D, 0.0D

GFORCE/11010, I = 11010, JFLOAT = 101, RM = 11010,
FUNCTION=USER(1,1,0,11010)
```

This pair of statements applies one aerodynamic force vector to marker 11010 at a location that is 0.74 length units from the origin of the local part reference frame of blade part number 10000. The blade is rotating clockwise when viewed looking downwind. The x (spanwise) component of the applied force is always zero. The blade element part is pitched (and/or twisted) in the ADAMS data set, and the GFORCE marker is parallel to the part.

The arguments of the GFORCE USER function are all integer values that are defined as follows:

Argument #	Definition
1	The blade number (1, 2, or 3)
2	The blade element number (1 through n, n≤20, with 1 being inboard and n being outboard)
3	Value of the VARVAL identifier that determines the aileron angle setting. When no power control is used, enter 0. When power control is used, enter the proper VARVAL index. This is a new, undocumented aspect of AeroDyn that is still under development. Users not wishing to use the control capabilities of ADAMS should always enter 0.
4	The ID number of the marker to which this aerodynamic force is applied (must be the same value as I and RM)

The blade element number provides the link between the ADAMS data set and the YAWDYN.IPT file. The first blade element in the YAWDYN.IPT file is element #1. There can be up to 20 blade elements for each blade and each blade must have the same element layout (location, chord, and airfoil). The pitch angle can be different for each blade to simulate pitch imbalance. Though the numbering of the aerodynamic force markers is arbitrary, we strongly suggest you use a consistent numbering system. The system we have used seems to work well. The inboard element of blade one is numbered 11010, the second element is 11020, the tenth element is 11100, etc. The inboard elements of blades two and three are 21010 and 31010 respectively. If you choose to use the REQSUB subroutine that is included as a sample, then the AERO marker numbers must match the values used in the REQSUB.

The AeroDyn software produces an output file that is useful for checking model inputs. The file is named GFOSUB.OPT. It echoes many input parameters and includes a table of blade element data that can be used to verify the location and orientation of each of the AERO markers at time=0. The blade element pitch is calculated by ADAMS in the same manner that is used during the simulation. This provides a thorough check of the blade marker orientations and locations.

14.3 Floating ground markers, Suggested IDs = 101-120, 201-220, 301-320

These markers are required for the GFORCE aerodynamic forces. One marker is required for each GFORCE. Each marker must be attached to ground and designated a floating marker. The orientation and location of the markers are arbitrary. Thus a typical ADAMS data set line would be:

```
MARKER/101, PART = 1, FLOATING
```

The markers associated with the blade elements on blade #1 have suggested numbers 101, 102, 103, up to the number of forces used on that blade. The markers for the aerodynamic forces on blade #2 are numbered 201,202, etc. These marker numbers are not required, but selection of these numbers will aid others users of the data sets.

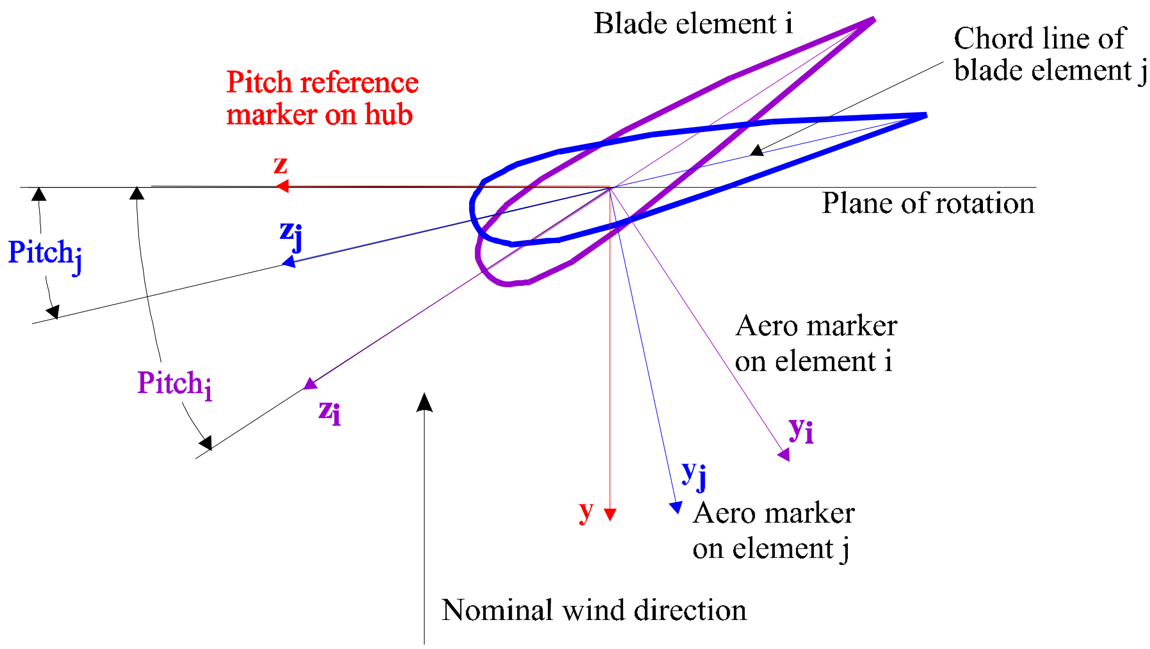


Figure 14.2 Definitions of aerodynamic pitch. The pitch angle determined by ADAMS for each blade element. The angles are shown in their positive sense. The **y** and **z** coordinates show the GFORCE marker orientations for two different blade elements i and j

14.4 Tower marker, ID = 1010

A marker must be attached to the topmost tower part and oriented with its Z-axis vertical downward as shown in Figure 14.1. This marker id is called ITOWER in the subroutines. This marker is used in conjunction with the yaw bearing marker (IYBRG=2010) to establish the yaw angle of the rotor.

If the tower experiences torsional deflections (relative to ground), the yaw angle will err by an amount equal to the torsional motion. This is a result of defining the yaw angle with respect to the tower top rather than with respect to ground (as should be done to more precisely model the aerodynamics of the rotor). It is expected that this error will be negligible for most towers, but the user should be aware of this limitation.

14.5 Yaw Bearing marker, ID = 2010

This marker must coincide with the tower marker, but it is attached to the nacelle part. The yaw angle of the rotor is the angle of rotation about the Z-axis of marker 2010 relative to marker 1010. This angle is used in the aerodynamics calculations. It is suggested that markers 2010 and 1010 be the same markers that establish the yaw bearing. It is essential that structural deflections of any kind not cause relative motion between these markers that could introduce any artificial rotor yaw. The simplest way to ensure this is to connect the parts upon which these markers are placed with a revolute joint, and to place the markers at the location of the revolute joint. This allows only one rotational degree of freedom.

14.6 Nacelle marker, ID = 2050

This marker is attached to the nacelle and must be located on the low-speed shaft (LSS or main) shaft centerline. It is used in conjunction with markers on the low-speed shaft to obtain blade azimuth angle, rotor angular velocity, and the radial position of each blade element. Since the instantaneous radial position of each blade element is determined by ADAMS as the radial distance between the AERO marker and the nacelle marker, it is imperative that marker 2050 be located on the LSS centerline. The marker is oriented such that the Z-axis is parallel to the shaft axis of rotation and x is nominally vertical downward. If the rotor axis is tilted, the x-axis will deviate from vertical by an amount equal to the tilt angle. (And the tilt angle must be entered in the YAWDYN.IPT file.)

14.7 Low-speed shaft markers, ID = 3051, 3052, 3053

One marker is required on the LSS for each blade. Each marker has its Z-axis parallel to the shaft axis of rotation and its X-axis directed radially outward in the plane formed by the blade span and the low-speed shaft. Thus, each marker has a Z-axis which is parallel to the z axis of marker 2050. The angle between the x axes of marker 2050 and 305n is the azimuth angle of blade n. Zero azimuth is defined when the blade is at the 6:00 position.

If the LSS is composed of beam or field elements, it is important that these LSS markers are located at the revolute joint that supports the LSS. If the markers are located at a point on the part that is not constrained to pure rotation (relative to the nacelle) then the Euler angles that are returned by ADAMS will give misleading information. This results from the fact that an infinitesimal amount of bending in the LSS will tilt the z-axis of these markers relative to the z-axis of the nacelle marker. When this happens the Euler angles between these markers, though correct, will not yield the blade azimuth angle as required by the GFOSUB.

14.8 Pitch reference markers, ID = 4191, 4291, 4391

A marker is required on the hub at the root of each blade. The markers are used as a reference from which the pitch angle of any blade element can be determined. Each marker must be oriented such that its z-axis is in the plane of rotation and its x-axis is parallel to the x-axes of the AERO markers. Thus, the x-axes of the pitch reference markers will project along the blade span, outwards for a rotor that spins clockwise when looking downwind and inwards for a rotor that spins counterclockwise. The markers will also be coned to match the rotor coning.

15.0 Adams Sensor Statement

ADAMS integrates the equations of motion using a scheme which adjusts the size of the time-step to achieve the desired solution accuracy. When a time step fails to meet convergence criteria, the integrator may step backwards in time and try a smaller step. Thus, the aerodynamics routines must be informed when a successful forward time step has occurred, and they must not update “old” values until a successful time step has been achieved. The SENSUB user-written subroutine provides a convenient method for determining when a successful time step has been completed. In the current version of the subroutines it also sets values of various time-tracking variables.

To invoke the SENSUB subroutine, a SENSOR statement <u>MUST</u> appear in your ADAMS data set. Without this statement, the time-based operations such as dynamic stall calculations and inflow averaging cannot be completed correctly. The subroutines will <u>not</u> always generate an error message if the SENSOR is not present. Therefore, it is essential that users take care to verify the operation of the SENSOR in their model. (Starting in version 10.0 there is one segment of code that tests for the presence of a working SENSOR statement. But this test will not work for very short simulations and has not been thoroughly tested and found reliable in other conditions.)

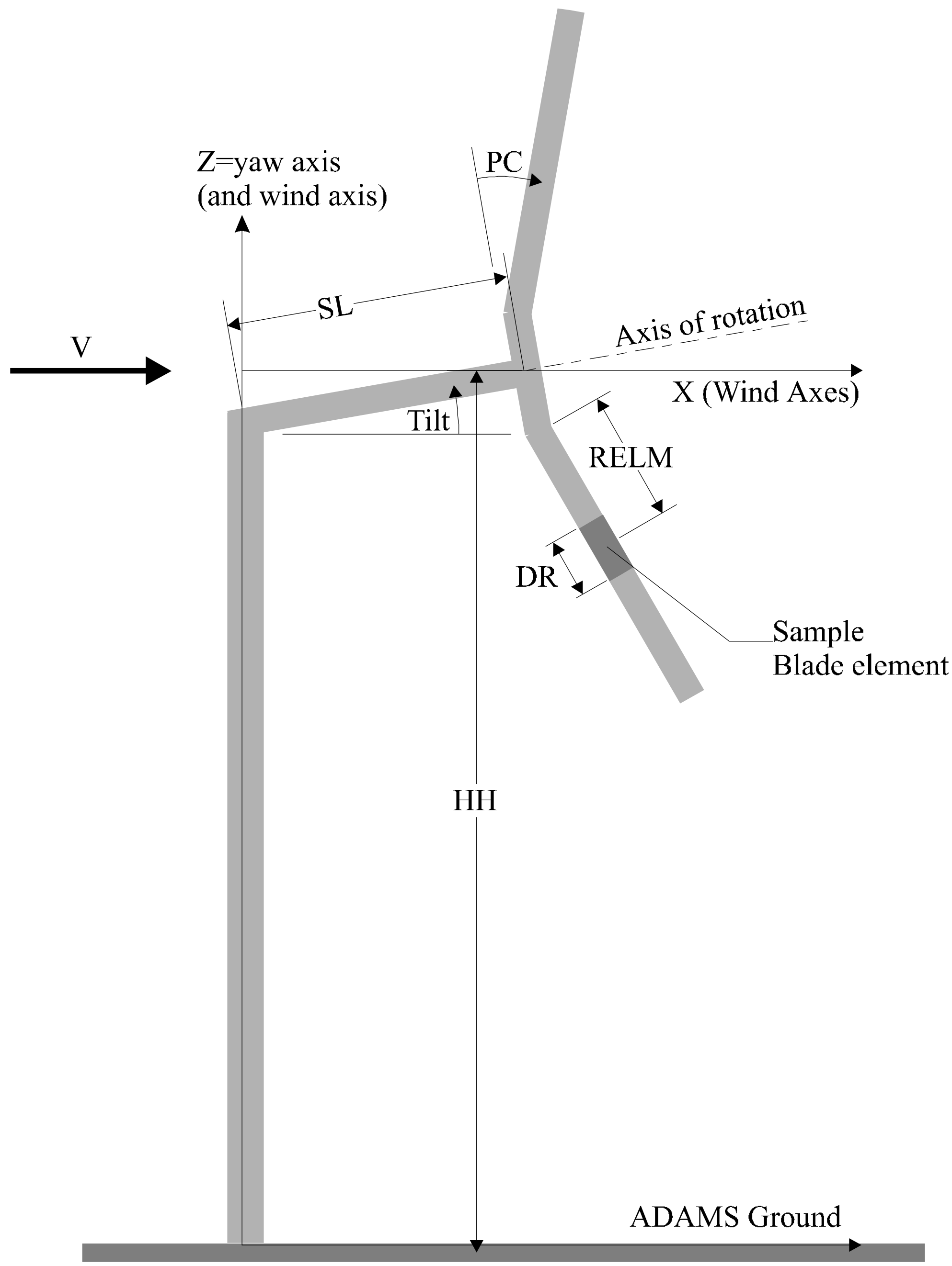


Figure 15.1 Coordinate systems and nomenclature used for the YAWDYN.IPT file parameters. The rotor is shown at zero yaw angle (with the shaft axis of rotation in the X-Z plane). The parameters are labeled with the variable names used in the YAWDYN.IPT file

A typical SENSOR statement appears below. The USER function must be specified. The other parameters in this example are used to avoid a bug that generates an incorrect error message when the data set is opened in ADAMS/VIEW version 6.0. They serve no other purpose and are not required if you are not using VIEW.

```
SENSOR/11111, VALUE=1., EQ, ERROR=1.0E-03
,  HALT, PRINT, YYDUMP
,  FUNCTION=USER(1.0)
```

16.0 Restrictions

The current version of the subroutines has the following restrictions. Our plan is to eliminate these restrictions over time. But this will require significant changes in the code and will not be done until we have validated the basic operation more fully.

1) The yaw angle is used to estimate the blade position in the skewed wake calculation. It is also used to calculate wind components relative to the rotor plane. This means ADAMS must have the yaw marker and tower-top marker shown in Figure 14.1.

2) The restriction on live twist in earlier versions of the routines has been eliminated. The routines now account for the live pitch of the blade whether the pitch is changing due to elastic twist or control inputs. However, the pitch angle is no longer obtained from the YAWDYN.IPT file or the YAWDYN.WND file. This makes it impossible to specify time-varying pitch using either of these files. If you want to specify a time-history of pitch angle you must use statements in the ADAMS data set to generate the desired motion.

17.0 The Request Subroutine (REQSUB.FOR)

The distribution disc contains a sample "Request" subroutine that is useful for obtaining outputs from ADAMS that are in a form that is relatively easy to use and customize. We expect that each user will want to generate outputs that are unique and specific to the particular model. For example, there might be reason for particular interest in the internal bending moments at a specific blade station. The request subroutine can be used to determine those loads of interest and send them to an output file. The sample subroutine obtains blade root moments and a variety of system parameters such as power output, yaw load, simulated time, blade azimuth and aerodynamic data. Any data that can be obtained from SYSARY or SYSFNC (or INFARY and INFFNC) subroutines in ADAMS can easily be accessed using the Request subroutine.

The data files are written in ASCII format with tab characters delimiting the columns. This makes it easy to import the results into a plotting or spreadsheet application for further processing or display.

This subroutine is invoked by including a statement of the following form in the ADAMS data set and linking the compiled REQSUB.OBJ file with ADAMS:

```
REQUEST/11111, FUNCTION = USER(1,0,0,0,0)
```

A separate REQUEST statement can be used for each blade, if desired. Each request will generate a separate data file named REQSUBn.PLT, where n is the blade number. The arguments of the USER function are defined in Table 17.1.

In the example statement given above, the subroutine will create a file named REQSUB1.PLT containing the simulated time series outputs including time, azimuth angle, flap moment, flap angle, yaw angle, yaw moment, and rotor power for blade #1 in a flapping hinge model. The first line of the data file contains column headings, also separated by tab characters.

NREL has released a general purpose post-processor (GPP) that can be used to extract results from ADAMS output files. This permits use of standard REQUEST statements in your ADAMS data set and eliminates the need for the REQSUB subroutine. It also provides capability for in-depth statistical analysis of the simulation results. NREL can provide information about the availability and use of GPP.

18.0 Questions And Answers

This section contains miscellaneous hints and tips for use of ADAMS with the wind turbine aerodynamics subroutines.

Table 17.1 - Arguments of the User-Written Request Function

Argument	Definition
1	The blade number (1, 2, or 3) for which output will be generated
2	No longer used (enter 0).
3	No longer used (enter 0).
4	An integer that identifies the type of rotor that is being modeled. 1 = Flapping hinge model 2 = Teetering rotor model Other values can be added for other turbine systems
5	No longer used (enter 0). In earlier versions of the code this parameter controlled creation of the AELEMENT.PLT file. This function is now controlled by PRINT/NOPRINT flag in the YAWDYN.IPT file.

18.1 Units

ADAMS is quite versatile in the units it allows. But the aerodynamics subroutines can be used with SI or English units only, unless you want to change a few lines in the source code.

If you are using standard SI units {Newton, meter, kg, sec}, then enter air density in kg/m^3 (e.g. 1.2 kg/m^3). If you are using English units {pound force, ft, slug, sec}, then enter air density in slug/ft^3 (e.g. 0.00238 slug/ft^3).

If you want to use units other than SI or English, be sure the air density (in the YAWDYN.IPT file) is selected to give a correct force unit. The aerodynamic force is calculated from $1/2C_L\rho AV^2$. The velocity V will be in the units selected for ADAMS. For instance, if you are using {pound force, inch, pound mass, sec} units, the velocity will be in inches/sec. You will have entered the airfoil chord and element length in inches. So the air density must be selected to give pounds force when multiplied by an area in square inches and a velocity squared in (inches/sec)^2. You must also go into the AEROSUBS.FOR source file and change the values of GRAV (acceleration due to gravity) and AS (speed of sound) to match the units system you are using. The use of alternate units is not recommended unless you are very familiar with the subroutines.

18.2 Time steps for ADAMS integration

We typically use 120 time steps per second when running ADAMS. Of course, the time step and the error tolerance for the integrator work together to determine the efficiency and accuracy of the solution. We have found that a very tight tolerance does not always lead to a more accurate solution, and it will slow the simulation considerably. A value of approximately 0.01 generally seems adequate. These values may not be appropriate for your model however, so we recommend you try a few different values and verify that your results do not depend strongly upon the size of the time step or the error tolerance. Recent models have been forced to use time steps as small as 0.002 sec. We have also found that run time is much shorter if care is taken to be certain that the output time step is an integer multiple of the HMAX integration time step. This prevents needless interpolation to generate the requested output step.

18.3 Startup problems and solutions.

We have tried a variety of ways to specify the initial conditions of a rotor. All of them introduce some startup transients in the simulation. The problem is analogous to finding the “trim” solution in any rotor dynamics problem. Some of the simulations we run are at constant rotor rpm with steady winds, and we simply specify a motion for the rotor (i.e. Motion = Omega * Time). This motion has infinite acceleration at time=0, so it generates large loads throughout the system at startup. The transient normally damps out in the first five seconds, so we just run longer simulations and use the results after all transients have disappeared.

We have tried smoother starts of the rotor speed, using polynomial and exponential motion functions. This seems to help when tower degrees of freedom are active. Abrupt rotor acceleration causes lateral tower bending that damps out slowly in the Combined Experiment model. When the tower is rigid, a smooth start takes about the same time as an abrupt start to reach a steady-state solution.

We have also applied extra damping at the beginning of a run to reduce the time required to reach a steady solution. For example, on the flapping hinge model of the blade an SFORCE is applied with a high damping value for the first few seconds and zero for later time. This does give smoother starts and avoids some convergence problems if the rotor has low aerodynamic damping.

When you simulate free yaw motion, it may be important to have the startup transients die out before the free yaw begins. To simulate this we have used statements to ACTIVATE and DEACTIVATE joints or other dataset elements in our command files. The technique we use most often to start a free-yaw simulation is to define two yaw joints in your dataset, one a revolute and the other a fixed joint at the same location. In your ADAMS command file you first deactivate the revolute joint and run a simulation for about five seconds. This allows the rotor to establish a “trim solution” in fixed yaw. You then deactivate the fixed joint and activate the revolute joint and run the simulation for the desired time of free yaw motion. See the ADAMS manuals for more information on this powerful and versatile technique.

You can use the wind to start the rotor slowly if the blade pitch is set for startup. The aerodynamics routines will work if the rotor is at zero rpm. This gives a smooth start, but it is a slow start and it requires additional logic to control the rotor speed as you reach synchronous speed. We have not explored this option in depth.

The following is an excerpt from an ADAMS/Wind Energy newsletter of August 1994. It offers additional suggestions on starting and running a successful simulation:

“Everyone who has run complex turbine models in ADAMS knows it is not at all trivial to successfully run long simulations. One of the more persistent and troublesome problems is starting a simulation smoothly. Generally a simulation begins with the entire turbine structure at rest, in the “as-built” configuration. If steady wind and control conditions are applied it will take some time for the model to come to equilibrium, or find a “trim” solution. Startup transients can be extreme, resulting in unstable motion and failure of ADAMS to successfully integrate the equations of motion. It is impossible to specify all the necessary initial conditions or to run a trim solution like YawDyn and FLAP. So you have to get the model to run long enough for all the transients to damp out, then you can begin the real simulation.

“An extreme example is starting an elastic (BEAM) model with a MOTION statement such as motion = omega*time. This imposes infinite acceleration at time = 0, and the resulting loads will probably wrap the blades around the low-speed shaft.

“Everyone I talk with about this is still struggling with the problem, because it changes slightly with each model. But here are some collected ideas and tips that may help you smoothly start a simulation without taking forever to have a “wind start”.

“1) Avoid step changes! Generator voltage can be specified with a STEP function to achieve a smooth rise in torque. Wind speed can be linearly increased from zero using a very short YAWDYN.WND file (see the User’s Guide). It is important for a static solution and for a smooth dynamic startup that the wind speed start at zero and increase to the desired value over a few tenths of seconds. For example, these first two columns from a .WND file will produce a linear ramp to a constant value. AeroDyn applies a wind that is linearly interpolated between the nearest values in the table, and the time steps in the table need <u>not</u> be equally spaced.

Time	Wind Speed
0.0	0.0
0.2	40.0
100.	40.0

“2) The typical model for an induction generator uses Thevenin’s equation. If you use ADAMS WT, this is the equation that you will probably be using. This equation is very well behaved when the generator is near synchronous rpm. However, during startup, there is a range of rpm’s over which the generator has negative damping. This occurs when the slope of the torque-speed curve is positive. I have seen this drive many models unstable as the rpm slowly increases during a startup. We have used a couple of tricks to overcome this difficulty.

“The first method is to apply a constant torque to the rotor (applied with a STEP function to avoid those abrupt changes) during the first few seconds of the simulation. This will accelerate the rotor through the unstable range quickly and avoid a serious problem. Some experimentation is required to find the best duration and magnitude of this torque.

“Another method seems to work well and require less experimentation. It requires a bit more code in your dataset. This method modifies the generator equation to apply a constant torque when the speed is below a specified value. These lines were taken from a model of a 250 kW machine we are analyzing at the University of Utah:

```
!                            adams_view_name='mot_gen'
SFORCE/1
, ROTATIONAL
, I = 216121
, J = 216122
, FUNCTION = if(varval(3)-varval(12):
,varval(10),
,varval(11),
,step(time,0.,0.,2.0,varval(11)))
!                           adams_view_name='line_volts'
VARIABLE/1
, FUNCTION = 240
!
adams_view_name='omega_desired'
VARIABLE/2
, IC = 5.235987756
, FUNCTION = 50*PI/30.0
!
adams_view_name='slip'
VARIABLE/3
, FUNCTION = 1-ABS(WZ(216121,216122,216122)/VARVAL(2))
!
adams_view_name='torq_curve'
VARIABLE/10
, FUNCTION = -2.5*varval(1)*varval(3)
,/(0.000157+0.00106*varval(3)+0.0243*varval(3)**2)
!
adams_view_name='torq_start'
VARIABLE/11
, FUNCTION = -2.5*varval(1)*varval(12)
```

```
,/(0.000157+0.00106*varval(12)+0.0243*varval(12)**2)
!
adams_view_name='slip_start'
VARIABLE/12
, FUNCTION = 0.04
```

"In this example the generator slip VARVAL(3) is monitored. If the slip is greater than 4% (VARVAL(12)), then the torque that would be calculated with 4% slip is applied (VARVAL(11)). This constant torque is actually applied with a two second step function to smooth the startup. When the slip becomes smaller than 4%, then the normal equation is applied. Note that the two torque speed equations (variables 10 and 11) are the same except that one uses the actual slip while the other uses 4% slip.

"3) Quite often it will help to precede a dynamic solution with a STATIC solution. This generally should be done with zero wind by using the .WND file. This seems to help by letting the structure find its equilibrium position under the influence of gravity alone.

"4) Additional damping or forces can be applied during startup to help get through difficult resonant speeds or other transients. Chapter 3 of the Solver manual discusses commands that can be used to change the properties of a model during the simulation. These commands are inserted into your command file (.acf file) rather than your dataset. For example, you might have a beam with a CRATIO of 0.001. You could specify a beam in your dataset with a CRATIO of 0.05. Then you could have a command file containing these lines:

SIMULATE/DYNAMIC, END=2.0, STEPS=100
BEAM/100, CRATIO=0.001
SIMULATE/DYNAMIC, END=10.0, STEPS=400

"This would run the first two seconds of the simulation with increased damping, then complete the ten second simulation with the actual value.

"Most parameters of a simulation can be changed "on-the-fly" in this way. We have modified spring-damper constants and also changed integrator parameters such as ERROR using commands. We have also activated and deactivated joints in this manner. Free-yaw machines seem to have particular difficulty with startups. They can be locked in yaw by starting with a FIXED joint, then changing to a REVOLUTE after the simulation has started.

"5) Finally, if you find yourself running long simulations only to find a problem occurring near the end of your run, you can save a lot of time and frustration by using the SAVE and RELOAD commands. Recently I ran a simulation with turbulent winds that ran fine for about 90 seconds of simulated time. Then it hit turbulence that caused the integrator to fail. I got past the problem by adjusting time steps and error tolerances for a short time around the 90 sec. point. The trial and error experimentation to find new integrator parameters was made a lot easier by SAVE and RELOAD. I ran a simulation for 89 sec, saved it, reloaded, changed the parameters, and continued. When I wanted to try new parameters, I just reloaded the successful 89 sec simulation and then tried the new values.

"Oddly, the SAVE and RELOAD technique seems to be able to stabilize a structure if it is beginning to go unstable. Dean Davis has used this method with the ESI-80 model. During startup it begins to diverge, with large teeter amplitudes and rates. He saves the simulation and when he reloads it the oscillations damp out. We believe this is a result of re-initializing the aerodynamics calculations. If the blade flap velocities become unrealistically high, the angle of attack changes rapidly and the dynamic stall calculations go outside the range for which they were intended. (We've seen flap velocities of hundreds of meters per second in an unstable model.) Restarting the aerodynamics calculations clears the angle of attack history and may help stabilize the model."

We are still looking for ways to improve the startup conditions. If you have any suggestions, please let us know.

18.4 Comment lines in ADAMS.

ADAMS has two ways to mark a comment line in the data set. You can place an exclamation mark in the first column or you can leave the first five (or more) columns blank. This second method has given us some surprises. We have spent too much time debugging data sets only to find a line that appeared to have no errors was ignored because it had five or more leading spaces.

18.5 Products of inertia.

Alan Wright of NREL pointed out that ADAMS solutions seem to proceed more smoothly when the PARTS have non-zero products of inertia (terms such as I_{xy}). A teetering rotor model was developed with all the products of inertia set to zero. When it ran, the integrator occasionally had to rotate parts to avoid singularities. When all of the product of inertia terms were set to non-zero, but negligibly small values, the part rotations were avoided with a subsequent saving of CPU time and no change in the final results. This is not fully understood, but it seems to help the simulations.

18.6 Additional Debugging Techniques.

It is not uncommon to have problems running a new model because of marker misalignment, pitch angle errors, or similar simple but fatal errors. These are some ideas you might use to isolate and identify the problem.

1. Examine the message (.msg) and GFOSUB.OPT files very closely. Errors or warnings generated by the aerodynamics routines are written to the screen and to the .msg file. It is easy to miss a warning written to the screen when it is scrolling, so the .msg provides a permanent record. The GFOSUB.OPT file contains information about your inputs. The blade element pitch angles are calculated by ADAMS using the same markers and methods that are employed in the aerodynamics calculations. This file is the key to determining whether your aerodynamic force markers are located and oriented correctly.

2. Turn off the wake calculations, and turn on the dynamic stall option. The wake calculations are time consuming and can increase the likelihood of numerical problems. If you are having problems running a new model, you can set the WAKE flag to NONE in the YAWDYN.IPT file to speed the calculations. Without the correct induced velocities your loads and power output will not be correct, but they should be adequate for debugging. (Be sure to turn the wake calculations back on when you finish debugging the model.)

   Dynamic stall usually increases the aerodynamic damping and stabilizes the models. When you are calculating the induced velocities, the dynamic inflow will also tend to provide aerodynamic damping. Since the aerodynamic damping is the largest contribution to total damping of a blade, these increases can be significant. Run your model in low winds to increase the aerodynamic damping. All of these steps will help get your model through the startup transients. After the model is debugged by running conditions with high damping, you should have little trouble running it in conditions with lighter damping.

3. Turn on the AELEMENT.PLT file option in the request subroutine (see section 17). Examination of the wind velocities and aerodynamics parameters in this file may give insight to problems with the model.

19.0 SAMPLE ADAMS DATA SET

A sample ADAMS data set is included on the distribution disk. It may be helpful to examine this model to see some of the details that are not covered in this manual.

References

Hansen, A. C., 1992, "Yaw Dynamics of Horizontal Axis Wind Turbines: Final Report." National Renewable Energy Laboratory, NREL Technical Report, TP 442-4822.

Hansen, A. C., C. P. Butterfield and X. Cui, 1990, "Yaw Loads and Motions of a Horizontal Axis Wind Turbine." Journal of Solar Energy Engineering. Vol. 112, No. 4. pp. 310-314.

Hansen, A. C. and X. Cui, 1989, "Analyses and Observations of Wind Turbine Yaw Dynamics." ASME Journal of Solar Energy Engineering. Vol. 111, No. 4. pp. 367-371.

Hansen, A. C. and A. D. Wright, 1991, "A Comparison of Combined Experiment Flap Load Predictions by the FLAP and YawDyn Codes." Tenth ASME Wind Energy Symposium, Houston, TX.

Hansen, A. C. 1993. "Structural Response to Unsteady and Stall Aerodynamics." 1993 DOE Wind Energy Program Review Meeting, Golden, CO.

Hansen, A. C., Butterfield, C. P., 1993. Aerodynamics of Horizontal-Axis Wind Turbines. Annual Review of Fluid Mechanics, 1993. Palo Alto, Annual Reviews, Inc.

Laino, D., Butterfield, C. P. 1994. "Using YawDyn to Model Turbines with Aerodynamic Control Systems." ASME Wind Energy Conference, New Orleans, LA.

Leishman, J. G., Beddoes, T. S., 1989. A Semi-Empirical Model for Dynamic Stall. Journal of the American Helicopter Society. 34(3): 3-17.

Malcolm, D. J., Wright, A. D. 1994. "The Use of ADAMS to Model the AWT-26 Prototype." ASME Wind Energy Symposium, New Orleans.

Peters, D. A., HaQuang, N., 1988. Dynamic Inflow for Practical Applications. Journal of the American Helicopter Society. 33(4): 64-68.

Appendix A. Tower Shadow Model

The tower shadow model has been improved from the model described in the references at the end of this guide. A schematic of the tower wake is shown in Figure A1. The wake is symmetric about its centerline. It is assumed to align with the instantaneous horizontal wind vector. (This assumption will be improved in a future revision by aligning the wind with a short-term average wind direction.) The velocity deficit in the wake is of the form

$$\text{deficit} = \begin{cases} u_1 \cos^2(\dfrac{\pi d}{2b}) & |d| \le b \\ 0 & |d| > b \end{cases}$$

and the horizontal components of wind speed are reduced by the deficit fraction:

$$VX = (1 - \text{deficit})VX_\infty$$
$$VY = (1 - \text{deficit})VY_\infty$$

Here b is the wake half-width, u_1 is the centerline deficit, and d is the perpendicular distance from the wake centerline to the point in question. The deficit is applied to the ambient horizontal wind rather than to the wind after it has been modified by the induced velocity of the rotor. Note the tower shadow has no effect on the vertical component of wind speed.

The width of the wake increases as the square root of the distance from the tower:

$$b = b_{ref}\sqrt{\frac{\iota}{\iota_{ref}}}$$

Where ι is the streamwise distance from the tower. The centerline deficit decays with streamwise distance according to the function

$$u_1 = u_{1ref}\sqrt{\frac{\iota_{ref}}{\iota}}$$

The reference length, ι_{ref}, is the distance from the yaw axis to the hub. This is shown as L_S in the sketch of the rotor geometry in the main body of this guide. In the program it is represented by the absolute value of the FORTRAN variable SL.

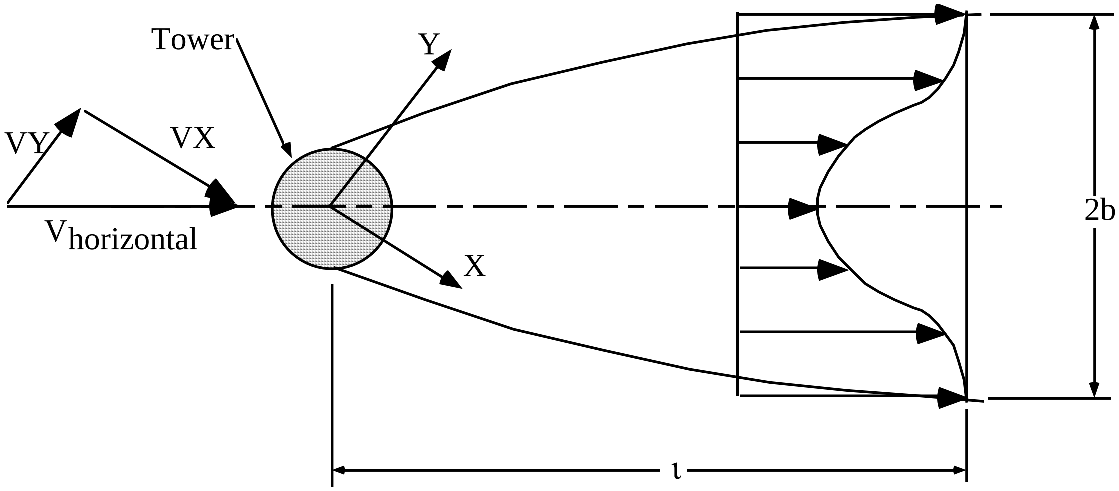


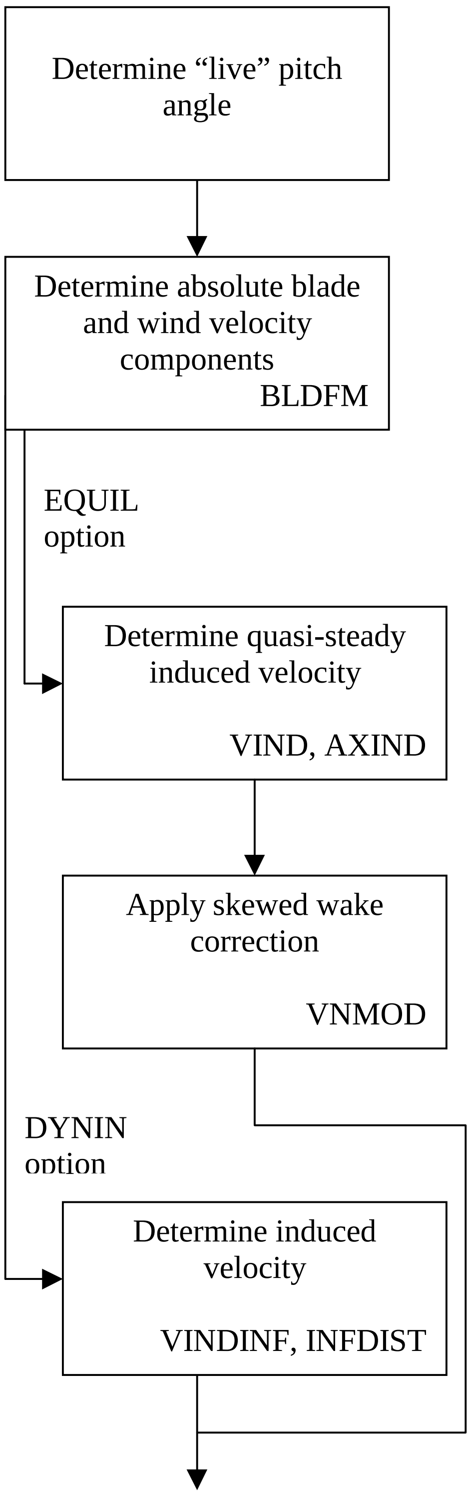
Figure A1. Schematic of the tower shadow model with a cross flow (VY). The tower wake decays in strength and grows in width as the distance from the tower, ι, increases. The strength and half-width are specified at a reference position, a distance L_S from the tower center.

Appendix B. Top-Level Flow Chart of the Aerodynamics Calculations

This appendix provides a simplified flow chart for the aerodynamics calculations. The purpose of the chart is to familiarize the user with the method that is used and some of the key assumptions. The chart does not map the flow of the entire program, nor does it use the format of traditional software flowcharts. The chart focuses on the operations rather than the code or subroutine structure. Each box of the chart does, however, indicate the name of the subroutine(s) in which the procedure is performed in the lower, right-hand corner of the box. This is intended to assist users who wish to examine the details within the subroutines.

The procedure that is shown is completed once for each blade element at each time step. Other subroutines, not shown in this chart, handle input and output of data and integration of the equations of motion.

Flow Chart of the Aerodynamics Calculations
and Major Assumptions in YawDyn
March, 1998



- All velocities expressed as components normal and tangential to the rotor plane
- Apply tower shadow, turbulence, wind shears
- Include all structural motions except pitch rate

EQUIL Option

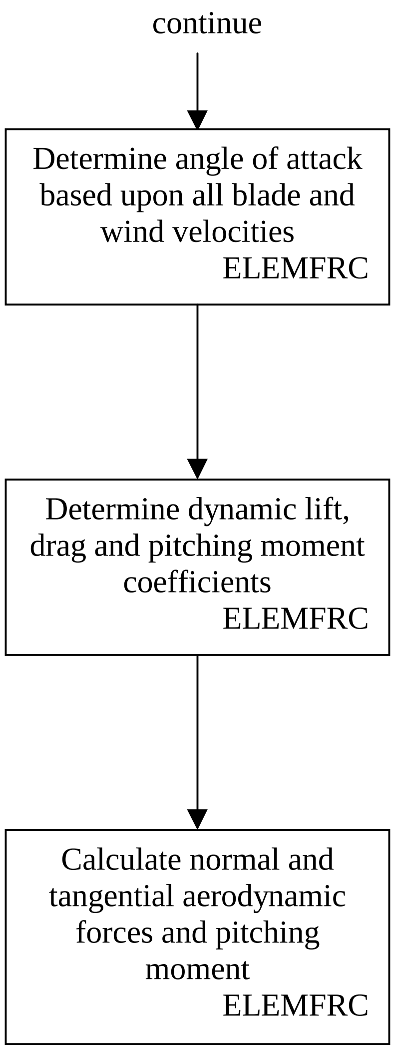
- Iterative procedure to determine induction factor
- Drag force excluded in momentum balance
- Prandtl tip-loss model included
- Static C_L is used in momentum balance

- Skewed wake correction based on dynamic inflow method of Pitt and Peters
- Applied in quasi-steady manner to the local induced velocity
- Uses average induced velocity to determine inflow angle to rotor

DYNIN Option

- Induced velocity calculated directly based on modified method of Pitt and Peters
- Skewed wake correction is inherent in the model
- Prandtl tip-loss model included

Aerodynamics Flow Chart (continued)



- All motions or wind components are considered in calculation of angle of attack

- Method of Beddoes and Leishman

- Based upon all blade and wind velocities
- Applied in the local coordinates of the element of the deformed blade

Appendix C. User's Guide to the FoilCheck Program

Introduction

Airfoil data are rarely available for angles of attack over the entire range of ±180°. This is unfortunate for the wind turbine designer, because wind turbine airfoils do operate over this entire range. The YawDyn program requires the user to provide airfoil data tables over the entire range so that it will be able to analyze any combination of wind speed, rotor speed, wind direction and yaw angle. If a table is not provided over the complete range, and an unusual angle of attack is encountered during the calculations, the program will terminate with an error message.

Fortunately, the aerodynamic characteristics of an airfoil generally become independent of the airfoil section shape for very high positive or negative angles of attack. This makes it possible to extrapolate from wind tunnel data (for the particular airfoil) to flat-plate characteristics for angles of attack near ±90°. The flat-plate lift and drag characteristics depend only upon the aspect ratio of the plate.

The FoilCheck program is a simple utility program that helps the user create an airfoil data table for YawDyn. It performs the following major functions:

1) Extrapolates airfoil data from a limited range of angles to the entire range of angles using flat-plate characteristics.

2) Calculates the parameters of the Beddoes dynamic stall model, based upon the static characteristics of the airfoil.

3) Writes an airfoil data file in the format required by YawDyn.

4) Writes an auxiliary file that can be examined to evaluate the "goodness" of the airfoil data file.

FoilCheck requires the user to start with a data file in the YawDyn format. All static airfoil characteristics of the airfoil that are known to the user must be contained in the input file. The static data <u>need not</u> cover the angles between ±180°. Dummy values for the dynamic stall characteristics must also be in the file as place holders.

<u>**CAUTION**</u>: The program <u>assists</u> the user in creating an airfoil data file. The process still requires accurate input data and judgment by the user. It does not completely automate the process of creating accurate data files. The user is prompted for inputs that require engineering judgment and, sometimes, a bit of guesswork. It is very important that the user check the resulting data file to be certain it is credible. This is one of the most important and difficult steps in creating an accurate YawDyn model of a turbine. We hope that FoilCheck eases the burden of creating the data files in the necessary format, but we know it cannot ease the burden of ensuring the data are accurate. The importance of accurate airfoil data cannot be overstressed. We encourage all users to devote considerable energy to locating airfoil data that is appropriate for the Reynolds number and surface roughness that will be seen on the turbine. The list of references at the end of this Appendix contains several sources of data for wind turbine airfoils over an extended range of angle of attack. Furthermore, extrapolation cannot be as accurate as test data. The accuracy of your turbine simulation is highly dependent upon accurate airfoil characteristics.

Method

FoilCheck uses a combination of wind-tunnel data, the Viterna equations for deep stall, and user experience to generate airfoil data for all angles from a limited set of measurements. The method is not proven, but it has been helpful to some users of YawDyn. So we decided to include the program in the YawDyn distribution.

Figure C1 shows lift and drag coefficients for an example airfoil. Letters A-G across the top of the plot show different regions of angle of attack. Region A is the location of the wind tunnel data for this airfoil. It is quite common to only have reliable data for angles between approximately 0° and 20°. All of the remaining regions are constructed from this data set, so it is clear that there is some hazard involved in the extrapolations.

Region B, from a point just beyond stall to 90°, is the region that the Viterna equations are applied in their original form. The equations are taken from a report by Viterna and Janetzke. (Note there is a typographical error in the equations in the report. The correct equations are given below.) Additional references are given at the end of this Appendix.

$$C_{D_{\max}} = 1.11 + 0.018AR \tag{1}$$

$$C_D = C_{D_{\max}} \sin^2 \alpha + B_2 \cos \alpha \tag{2}$$

where

$$B_2 = \frac{C_{D_s} - C_{D_{\max}} \sin^2 \alpha_s}{\cos \alpha_s} \tag{3}$$

and subscript s denotes the value at the stall angle (called the matching point in this User's Guide because it need not be exactly at stall). *AR* is the blade aspect ratio. The lift is given by

$$C_L = \frac{C_{D_{\max}}}{2} \sin 2\alpha + A_2 \frac{\cos^2 \alpha}{\sin \alpha} \tag{4}$$

where

$$A_2 = \left(C_{L_s} - C_{D_{\max}} \sin \alpha_s \cos \alpha_s\right) \frac{\sin \alpha_s}{\cos^2 \alpha_s} \tag{5}$$

These equations yield C_L=0 and C_D=C_{Dmax} at α=90°, and the stall (or matching point values) at α_s. Thus it is important to select α_s carefully.

In Regions C, D and E of Figure C1, values are obtained by scaling and reflecting the values from Region B. The reflections are evident from the figure. FoilCheck applies a scaling factor to C_L to account for the asymmetry of the airfoil. The scaling factor is 0.7. That is, all lift values are reduced by 30% from the values shown in Region B. Drag values are not scaled, just reflected. In regions F and G, linear interpolation is used to connect the various regions. C_L is forced to zero at α=±180°.

Pitching moment coefficients can also be extrapolated from tabular data. The extrapolation method is based upon the following assumptions:

1) The center of pressure moves to the midchord at α=90°. This implies CM at 90° is -$C_{Dmax/4}$.

2) The location of the center of pressure can be estimated using a Tan(α) function. The curve is fit between the value at 90° at the matching point, α_s. This assumption is not proven, but gives reasonable results in the few cases where data are available over the entire range of angles of attack. Better results will be obtained in this extrapolation if you have tabulated values for angles greater than 20°.

3) The CM curve is reflected to positive values for negative angles of attack.

4) Fixed values are used at angles near 180°. The pitching moment can be large for reversed flow over the airfoil. Some wind turbines have experienced structural failures in their pitch control system as a result of this type of flow in very high winds. We therefore elected to use large and constant values as follows: (-170°, 0.40), (180°, 0.0), (170°, -0.50). These are the largest (absolute) values seen in limited data that are available from the Ohio State University reports listed at the end of this appendix. Of course, if you have data for your airfoil, we suggest using that data instead of FoilCheck.

The program is written in modular form that we hope will be easy for others to understand. If other users can improve upon any of these assumptions, we hope they will modify FoilCheck accordingly, and inform us of what they find.

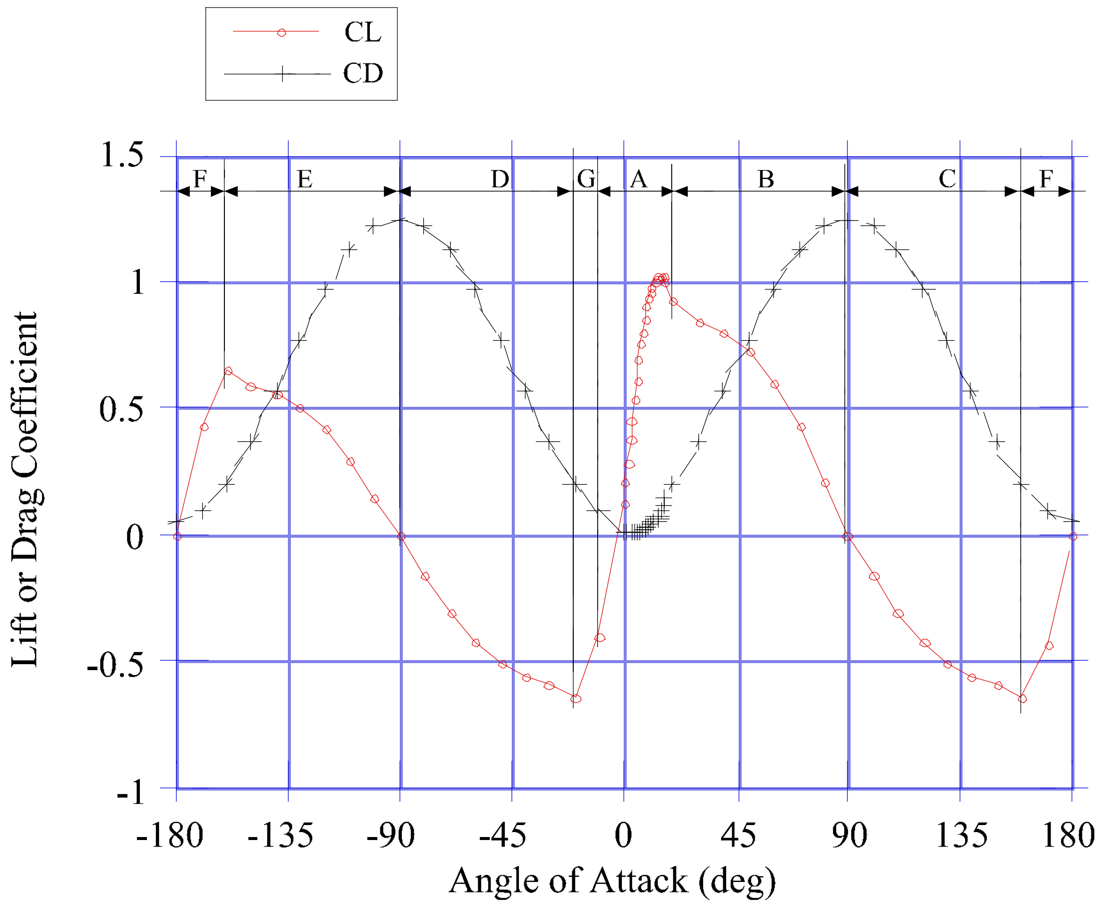


Figure C1. Lift and drag coefficients for a typical airfoil.

Installation

The source code for FoilCheck consists of three files: 1) FOILCHK.FOR contains all the source code main program and subroutines. 2 and 3) AERODYN.INC and BEDOES.INC are include files that must be present in the same directory when the program is compiled and linked. These are the same include files that are used with YawDyn.

The source code must be compiled and linked using any Fortran 90 compiler. (NOTE: If desired, a Fortran 77 compiler can be used by making adjustments to the AERODYN.INC file as described in the comments at the beginning of that file. This is the only file where Fortran 90 specific code exists in the FoilCheck code.)

Input data

FoilCheck requires two types of input. The data file read by the program must be in the same format as the airfoil data file used by YawDyn and AeroDyn. This format and file are described in the YawDyn User's Guide. The other type of input is interactive input from the keyboard during program execution. This is detailed later in this guide.

The format of the airfoil data file must match that of the YawDyn airfoil data file exactly. However all numerical values related to the dynamic stall characteristics need not be accurate. These values are found in lines 5 through 12 in the data file. (Lines 3 and 4 - the number of airfoil tables and the ID parameter – must be accurate.) Also, the static airfoil lift and drag table should only cover the range for which values are accurately known. It is not necessary to provide a table starting at -180° and ending at +180°. FoilCheck will read the static table provided and build the new table based upon these values. A sample input data file is shown in Table C1. Note the use of zeroes for all of the dynamic stall inputs and the relatively short range of angles of attack for the static lift and drag coefficient table.

Table C1 - Sample input airfoil file

```
S809 Airfoil, OSU data at Re=.75 Million, Clean roughness
NREL/TP-442-7817 Appendix B
   1        Number of airfoil tables in this file
      .00   Table ID parameter
     0.00   Stall angle (deg)
     0.00   Not used
     0.00   Not used
     0.00   Not used
     0.00   Zero lift angle of attack (deg)
     0.00   Cn slope for zero lift (dimensionless)
     0.00   Cn at stall value for positive angle of attack
     0.00   Cn at stall value for negative angle of attack
     0.00   Angle of attack for minimum CD (deg)
     0.00   Minimum CD value
   -20.10  -.560  .3027   .0612    Alpha, CL, CD, CM
   -18.10  -.670  .3069   .0904
   -16.10  -.790  .1928   .0293
   -14.20  -.840  .0898   -.0090
   -12.20  -.700  .0553   -.0045
   -10.10  -.630  .0390   -.0044
    -8.20  -.560  .0233   -.0051
    -6.10  -.640  .0131   0.0018
    -4.10  -.420  .0134   -.0216
    -2.10  -.210  .0119   -.0282
      .10   .050  .0122   -.0346
     2.00   .300  .0116   -.0405
     4.10   .540  .0144   -.0455
     6.20   .790  .0146   -.0507
     8.10   .900  .0162   -.0404
    10.20   .930  .0274   -.0321
    11.30   .920  .0303   -.0281
    12.10   .950  .0369   -.0284
    13.20   .990  .0509   -.0322
    14.20  1.010  .0648   -.0361
    15.30  1.020  .0776   -.0363
    16.30  1.000  .0917   -.0393
    17.10   .940  .0994   -.0398
    18.10   .850  .2306   -.0983
    19.10   .700  .3142   -.1242
    20.10   .660  .3186   -.1155
```

The interactive input is best described using an example. The left column of the table below is a copy of the prompts and user input to the screen during program execution. (The Screen display is shown in `Courier` font, user inputs are shown **bold**. Spacing and fonts have been altered, and borders have been drawn for ease of reading, otherwise the left column is a copy of the screen display.) The right column provides some additional description of the options.

Note that the program prompts has changed quite a bit with version 11.1 (the method of the program has not changed). Default values are offered for many of the dynamic stall parameters, as well as for all the "y/n" prompts. The choice in square brackets, "[ ]", is the default that is used if you simply hit the Enter key. Some prompts also offer a "?" option which will provide more help on that topic if selected.

Text appearing on computer display (both prompts and responses)	Explanation

`WELCOME TO FoilCheck for YawDyn 11.0 (26-Mar-1998)`
`==================================================`
`(Respond to the prompts to generate`
`airfoil data tables for YawDyn)`
 	
`Enter the name of the airfoil file that you wish`
`to examine (<80 characters)`
s809_cln.dat
	You are first prompted for the name of the file containing the airfoil data. Any name, including path, up to 80 characters is accepted.

`File "s809_cln.dat" found.`
 	
`Does this file contain aerodynamic pitching`
`moment coefficient data?`
`Y/[N] or ? >`**y**
	You must tell FoilCheck if your input data file contains pitching moment coefficients. If you wish to have your output file contain C_M values, then your input file must contain them.

`Finished reading airfoil data file` `"s809_cln.dat"`	The system responds by telling you it has read the airfoil file. If you get an error message check to be sure your file name and path are correct, then check the input file to be certain it has the correct number of lines and values.

`FoilCheck determined there are  2`
`airfoil tables in this file.`
`Enter the number of the airfoil table`
`that you wish to examine.`
1
	This prompt appears only if you have more than one airfoil table in your data file. FoilCheck can only output results for one table, but it can be any table from your input file.

`Do you want to calculate lift and drag`
`coefficients using the Viterna method?`
 	
`(You should answer yes unless you are starting`
`with a table that covers -180 < alpha < 180)`
`Y/[N] >`**y**
	If your airfoil table does not include angles over the entire range between ±180°, you can use the Viterna equations to generate static C_L and C_D values. 'Y' or 'y' responses will invoke the Viterna calculations.

`The blade aspect ratio (AR) is used to estimate`
`the maximum Cd for the airfoil (Cdmax)`
`using the equation Cdmax = 1.11 + 0.018*AR`
 	
`If you wish to enter a different Cdmax then`
`you should enter an aspect ratio of zero`
 	
`Enter the aspect ratio of the blade`
 	
11
	The Viterna method requires input of C_{Dmax}. You can either enter the aspect ratio, from which the maximum drag is calculated, or you can enter 0. You will then be prompted for a value of C_{Dmax}.

```
The lift and drag coefficients will be matched at
 the angle that you enter below and calculated
 for angles greater than the value you enter.

The angle should be at or above the stall angle.
 Normally, the largest angle in your table is best.

It should also be one of the values appearing
 in your original airfoil table.
     [Press ENTER to continue]

 Here is a list of tabulated values near stall
    ALPHA      CL        CD
     10.20     0.930    0.0274
     11.30     0.920    0.0303
     12.10     0.950    0.0369
     13.20     0.990    0.0509
     14.20     1.010    0.0648
     15.30     1.020    0.0776
     16.30     1.000    0.0917
     17.10     0.940    0.0994
     18.10     0.850    0.2306
     19.10     0.700    0.3142
     20.10     0.660    0.3186


 Enter the angle of attack you wish to use for
  the matching point (in degrees)
20.1
```

This is the most critical step in using the Viterna method. The equations fit a smooth curve between the matching point you enter here and finite-length flat plate characteristics at higher angles. The angle should be above stall, but generally not above approximately 30°. Angles between 15°-20° are most common. It is important to examine the resulting airfoil file after the program runs to be certain the curve is reasonable. But remember, FoilCheck is correcting for aspect ratio, so the results will not match 2-D data.

The program will list a table of values from which to choose.

You should enter a value from the input table. If you do not, you will be prompted for C_L, and C_D values for the angle you entered.

```
To calculate the new airfoil table, FoilCheck needs
   the matching point for the lower bound.

The lift and drag coefficients will be matched at
the angle that you enter below and calculated
for angles less than the value you enter.

A typical value will be near zero degrees.
(You may choose the smallest angle in your table.)

It should be one of the values appearing
in your original airfoil table.
     [Press ENTER to continue]


Here is a list of tabulated values near zero de-
grees
   ALPHA      CL        CD
   -18.10    -0.670    0.3069
   -16.10    -0.790    0.1928
   -14.20    -0.840    0.0898
   -12.20    -0.700    0.0553
   -10.10    -0.630    0.0390
    -8.20    -0.560    0.0233
    -6.10    -0.640    0.0131
    -4.10    -0.420    0.0134
    -2.10    -0.210    0.0119
     0.10     0.050    0.0122
     2.00     0.300    0.0116
     4.10     0.540    0.0144

Enter the lower-bound angle of attack (in deg.)
  The value must not be less than the lowest
  angle in the original airfoil table.
-18.1
```

A range is established by your last entry and this next entry (20.1° and -18.1° in this example). Airfoil values for angles outside this range are created by calculating, scaling and reflecting the values within the range as described in the text of this User's Guide.

A table of values near zero degrees will be listed. You should select one of these values.

Again, it is important to look at the final airfoil file to see if good choices were made while running the program. This judgment is one of the most difficult parts of the process. FoilCheck cannot assist your selection other than by listing candidate values from your input table.

```
The zero-lift pitching moment coeff. = -0.0334

  A new data table has been created.
  The number of points in the three intervals:
         AOA below the table range:           17
         AOA within the table range:          25
         AOA above the table range:           16
  Number of entries in the table =          58

     [Press ENTER to continue]
```

The program creates a new data table that contains your original values, plus values at 10° intervals over the entire range outside of your original table.

<pre>A first iteration for the CN slope was performed
  with the following results obtained:
  [Press ENTER to continue]
 	
 Minimum angle of attack for the interval = -2.100
 Maximum angle of attack for the interval =   6.200
 Number of points in the interval   = 5
 	
 CN Slope from linear least squares fit = 6.9071
 CN Slope read from the airfoil table   = 0.0000
 CN intercept from least squares fit   = 0.0450
 	
  ALPHA (DEG)   CN-TABLE   CN-CALCULATED   ERR
   -2.100   -0.210   -0.208   0.0021
    0.100   0.050   0.057   0.0070
    2.000   0.300   0.286   -0.0141
    4.100   0.540   0.539   -0.0004
    6.200   0.787   0.792   0.0054
 	
  Root-Mean-Square CN error from curve fit
    calculated over the specified interval
      RMS error = 0.0075
 	
 Do you want to do another CN slope? (Y/[N])
Y</pre>
	We have finished creation of the static airfoil table for all angles of attack. Now we turn to the dynamic stall parameters. First is the normal force slope. The program fits a least-squares line through the C_N values for a range of angles you specify. The slope required by the theory is the slope at C_N=0.
	
	You must reach a balance between having enough points in the curve fit to give confidence in the slope, and not increasing the range to large angles (large C_N). The program makes an initial attempt to find a slope spanning C_N = 0.0, with an RMS error < 0.01. The result is written to the screen.
	
	You can evaluate the fit and decide whether to try a different range or accept this range. While this first attempt usually provides useful results, in this example let's try a different range.

<pre>The program will calculate the CN slope
for the series consisting of all the data
points in the angle-of-attack range that you
specify in the next two lines of input
 	
Enter the minimum angle of attack (deg)
-3
Enter the maximum angle of attack (deg)
3
 	
Minimum angle of attack for the interval =   -3.000
Maximum angle of attack for the interval =   3.000
Number of points in the interval   =   3
 	
CN Slope from linear least squares fit =   7.1250
CN Slope read from the airfoil table   =   0.0000
CN intercept from least squares fit   =   0.0466
 	
 ALPHA (DEG)   CN-TABLE   CN-CALCULATED   ERR
  -2.100   -.210   -.214   -.0042
    .100   .050   .059   .0091
   2.000   .300   .295   -.0049
 	
 Root-Mean-Square CN error from curve fit
   calculated over the specified interval
RMS error =   .0064
 	
Do you want to do another CN slope? (Y/N)
n
 	
  A CN slope value of   7.124995   will be
   written to the new airfoil data file.
        [Press ENTER to continue]</pre>
	By trying a smaller range we get a smaller RMS error.
	
	The 'n' response tells the program to accept this last value of C_N slope and move on to the next set of questions.
	
	You can answer 'y' as many times as you like, until you get a result that is satisfactory. If you want to use one of the results you saw earlier, run that range of angles again.
	
	The CN-TABLE value is calculated from the C_L and C_L in your input table. CN-CALCULATED is the value calculated from the linear regression through the CN-TABLE values. ERR is the difference between the two.
	
	The zero-C_N angle of attack is also determined from the results of the linear regression.

<table>
<tr><td>
<pre>
Next you must enter the stall angle of attack.

  The following points are your table values
     at angles bracketing stall:

      ALPHA   CL
     10.20   0.930
     11.30   0.920
     12.10   0.950
     13.20   0.990
     14.20   1.010
     15.30   1.020
     16.30   1.000
     17.10   0.940
     18.10   0.850
     19.10   0.700
     20.10   0.660
  A stall angle of attack of   15.30000
   has been found by FoilCheck.

  Do you want to accept this value ([Y]/N)?
n

  Enter the angle of attack at stall (deg)
15.7
</pre>
</td><td>
FoilCheck is requesting the angle of attack for stall. To assist you, FoilCheck echoes your input table for angles near stall. FoilCheck also provides a default value for this angle of attack based on the angle of attack in the table with the maximum C_L. To accept this value enter “y” or simply press the Enter key.

If you wish to enter a different value than the default, enter “n”. You are then prompted to enter the stall angle. The value is entered in degrees. It does not have to equal one of the points from the input table.

FoilCheck calculates the C_N at this stall angle. The value is extrapolated from the linear C_N-curve slope found above. This has been found to yield better results than the static value from the table.
</td></tr>
<tr><td>
<pre>
 Now you must enter a value for CN at the
  stall point for negative angles of attack.

 The program will use a default value
  Equal to -0.8 if you would like.

 Do you want to accept this default value ([y]/n)?
Y
</pre>
</td><td>
Now we look at “negative stall”. If you know the value of C_N at stall for negative angles, reject the default and enter the actual value. Otherwise, accept the default value.
</td></tr>
<tr><td>
<pre>
Next you must enter the angle of attack for Cdmin.

 The following points are your table values
    at angles bracketing minimum CD:

     ALPHA   CD
    -8.20   0.0233
    -6.10   0.0131
    -4.10   0.0134
    -2.10   0.0119
     0.10   0.0122
     2.00   0.0116
     4.10   0.0144
     6.20   0.0146
     8.10   0.0162
    10.20   0.0274
    11.30   0.0303

 A minimum CD value of 0.0116
  At an angle of attack of   2.00 deg
  Has been found by FoilCheck.

 Do you want to accept this value ([Y]/N)?
N

  Enter the angle for minimum CD (deg)
1.5
</pre>
</td><td>
The final entry is the angle of attack at which C_D is a minimum (near zero degrees, not 180°). A default from this table is presented by FoilCheck. Enter “y” (or press the Enter key) to accept this default, or “n” to input your own choice.
If you reject the default, you are prompted to enter an angle. It need not be from the table that is input, but the table is echoed to the screen to help you select a value.
</td></tr>
</table>

`Finished. Three data files have been written`
`    foilnew.plt is the new airfoil data file`
`    foilchk.plt contains diagnostic data`
`    foilchk.opt is a copy of some screen output`
 	
`Press Enter to exit program`
	FoilCheck finishes by informing you of the files generated during the session.
	
	You can now press the Enter key to exit FoilCheck.

FoilCheck creates three new data files. FOILNEW.PLT is the new airfoil data file in the YawDyn format. You can edit the first two lines of the file to replace the generic TITLEs with meaningful notes regarding the file contents. This file contains only one airfoil table—for the table that was used as input. If you are using airfoil files with multiple data tables (such as aileron tables) you must cut and paste the multiple table file together from the many FOILNEW.PLT files that you will generate by running FoilCheck many times. FOILCHK.OPT contains the results of all of the CN slope calculations for review if desired. FOILCHK.PLT contains a number of calculated values for checking your results. It is a tab-delimited ASCII file suitable for import to a variety of graphics or spreadsheet programs for plotting. The column identification and description are shown in Table C2 below. We strongly suggest you examine this file closely to verify the accuracy of your airfoil tables as much as possible.

Table C2 - Column headings in the FOILCHK.PLT file.

Column Heading	Description
Alpha	Angle of attack in degrees, from -180 to +180.
CL	Static lift coefficient
CD	Drag coefficient
CM	Pitching moment coefficient
CN	Normal coefficient (static)
CT	Tangential (chordwise) coefficient (static)
LiftDragRatio	The Lift/Drag ratio. This can be used to check the results. Very high or very low maximum L/D, or sharp discontinuities in the values should be examined closely.
CLBeddoes	This is the value of CL calculated from the Beddoes dynamic stall parameters *for static conditions*. It should match the CL values of column 2 very closely. If it does not, there is an error in one of the Beddoes parameters.
CDBeddoes	The drag coefficient calculated from the Beddoes dynamic stall parameters. This should also match the CD values of column 3 very closely
FtbBeddoes	A parameter used in internal dynamic stall calculations. It represents the fraction of attached flow on the airfoil. Values range from zero to one. This is available for users who are familiar with the details of the Beddoes dynamic stall calculations.

References for Appendix C

Butterfield, C. P. 1989. "Aerodynamic Pressure and Flow-Visualization Measurement from a Rotating Wind Turbine Blade." Eighth ASME Wind Energy Symposium, Houston, TX.

Butterfield, C. P. 1989. "Three-Dimensional Airfoil Performance Measurements on a Rotating Wing." European Wind Energy Conference, Glasgow, Scotland.

Butterfield, C. P., Simms, D., Scott, G., Hansen, A. C. 1991. "Dynamic Stall on Wind Turbine Blades." Windpower '91, Palm Springs, CA.

Eggers, A. J., Digumarthi, R. V. 1992. "Approximate Scaling of Rotational Effects of Mean Aerodynamic Moments and Power Generated by the Combined Experiment Rotor Blades Operating in Deep-Stalled Flow." Eleventh ASME Wind Energy Symposium, Houston, TX.

Gormont, R. E., May, 1973. "A Mathematical Model of Unsteady Aerodynamics and Radial Flow for Application to Helicopter Rotors." U.S. Army Air Mobility Research and Development Laboratory, USAAMRDL Technical Report, 76-67.

Leishman, J. G., Beddoes, T. S., 1989. A Semi-Empirical Model for Dynamic Stall. Journal of the American Helicopter Society. 34(3): 3-17.

Miley, S. J., February, 1982. "A Catalog of Low Reynolds Number Airfoil Data for Wind Turbine Applications." Rockwell International Rocky Flats Plant, RFP-3387.

Roensten, G., Dahlberg, J. A., Meijer, S. 1989. "Pressure Measurements on a 5.35m HAWT in CARDC 12x16m Wind Tunnel Compared to Theoretical Pressure Distributions." European Wind Energy Conference, Glasgow, Scotland.

Snel, H., Houwind, R., 1993. "Section Prediction of 3-D Effects for Stalled Flow on rotating Blades and Comparison with Measurements." European Community Wind Energy Conference, Luebeck_Travemuende, Germany.

Tangler, J. L., Ostowari, C., June, 1991. "Horizontal Axis Wind Turbine Post Stall Airfoil Characteristics Synthesization." NREL, SERI/TP-257-4400.

Viterna, L. A., Corrigan, R. D. 1981. "Fixed-Pitch Rotor Performance of Large HAWTs." DOE/NASA Workshop on Large HAWTs, Cleveland, OH.

Viterna, L. A. and Janetzke, D. C., Theoretical and Experimental Power From Large Horizontal-Axis Wind Turbines. NASA TM-82944, September, 1982.

Sources of airfoil over an extended range of angle of attack:

Janiszewska, J. M., R. Reuss Ramsay, et al. (1996), Effects of Grit Roughness and Pitch Oscillations on the LS(1)-0417MOD Airfoil. National Renewable Energy Laboratory, NREL/TP-442-7819.

Pope, A. (1949). “The Forces and Moments Over an NACA 0015 Airfoil Through 180° Angle of Attack.” Aero Digest: p. 76.

Reuss Ramsay, R., M. J. Hoffman, et al. (1995), Effects of Grit Roughness and Pitch Oscillations on the S809 Airfoil. National Renewable Energy Laboratory, NREL/TP-442-7817.

Reuss Ramsay, R., M. J. Hoffman, et al. (1995), Effects of Surface Roughness and Vortex Generators on the LS(1)-0417MOD Airfoil. National Renewable Energy Laboratory, NREL/TP-442-6474.

Reuss Ramsay, R., M. J. Hoffman, et al. (1995), Effects of Surface Roughness and Vortex Generators on the NACA 4415 Airfoil. National Renewable Energy Laboratory, NREL/TP-442-6472.

Reuss Ramsay, R., M. J. Hoffman, et al. (1996), Effects of Grit Roughness and Pitch Oscillations on the LS(1)-0421MOD Airfoil. National Renewable Energy Laboratory, NREL/TP-442-6473.

Reuss Ramsay, R., M. J. Hoffman, et al. (1996), Effects of Grit Roughness and Pitch Oscillations on the S801 Airfoil. National Renewable Energy Laboratory, NREL/TP-442-7818.

Reuss Ramsay, R., M. J. Hoffman, et al. (1996), Effects of Grit Roughness and Pitch Oscillations on the S810 Airfoil. National Renewable Energy Laboratory, NREL/TP-442-7816.

Appendix D Sample Batch Files for ADAMS with User-Written Subroutines

We have found batch files are useful for compiling, linking and running user-written subroutines with ADAMS version 8.1 in Windows NT. This Appendix describes some of the more useful batch files we use at the University of Utah. The files may be useful to others either directly (if one wishes to use the same sub-directory names and organization) or indirectly, as a guide to the use of compiler options and the linking requirements. Of course, these examples will only work with a hard disc that has the same directory tree. But it is fairly simple to change the directory names in the batch files to make this system work with a different tree.

The batch files for compiling set the necessary compiler options for ADAMS. These are not the only (and probably not the best) options, but we have found that they work.

Both the MDI (C:\MDI81) and MS FORTRAN (C:\FPSNT) directories on our hard disc were created using the default names and structure during the INSTALL procedures for the two products. If you used different directory names or organization your batch files will have to be changed accordingly. The NT environment variables are established during the installation of ADAMS and FORTRAN. You should not need to alter these variables.

If you wish to compile, link and run routines with ADAMS you should take the following steps while operating in a Command (or DOS) window:

1) Compile each subroutine file using the Compile8.bat file. For example, type **compile8 GFOSUB.FOR** to compile the main GFOSUB routine. The GFOSUB.FOR, AEROSUBS.FOR, REQSUB.FOR, and SENSUB.FOR routines must be compiled with this command. The current version of ADAMS also has a UCONFG.FOR file that should be compiled and linked when running large models. We keep all of these source files in a directory called C:\ADAMS. This is also the directory in which the executable file (in our case called AERODYN.EXE) is stored. We call the Compile8.bat only when we are in the C:\ADAMS directory.

2) Link all the routines to ADAMS by typing **VLINK**. This invokes the ADAMS link routine and creates the executable file. We have to be in the C:\ADAMS directory when we execute this batch file.

3) Move to the directory that contains your data files (the .ADM file, the YAWDYN.IPT file, and the airfoil data files) and type **SIM** to run the SIM.BAT file. Before you can use SIM you must have a command file in the same directory (we call our SIM.ACF) that lists the ADAMS commands (see MDI manuals). A sample SIM.ACF file is listed below.

Compile8.bat Compile Microsoft FORTRAN Powerstation 32 for ADAMS version 8.1

```
fl32 /c /G5 /Op /4Yb %1
```

Vlink.bat Link FORTRAN for ADAMS version 8.1

```
mdi cr-u < Linkinpt.dat
```

Linkinpt.dat file used by Vlink.bat. Note this file contains a blank line before the last line.

```
gfosub.obj
aerosubs.obj
reqsub.obj
sensub.obj
aerodyn.exe
```

Sim.bat Run ADAMS analysis

```
del reqsub1.plt
del reqsub2.plt
mdi ru-u c:\adams\aerodyn.exe sim.acf
```

Sim.acf Sample ADAMS command file to run a simple YawDyn-equivalent model of the Combined Experiment Rotor named hinge.adm. Joints are changed to allow switching between fixed- and free-yaw simulation. (It is not necessary to have the activate and deactivate commands in the command file, they are included to illustrate the technique.)

```
hinge
hinge
deactivate/joint, id=2001
sim/dyn, end=4.69, steps=469
deactivate/joint, id=2000
activate/joint, id=2001
sim/dyn, end=40.0, steps=3531
stop
```

Appendix E Description of Dynamic Inflow Model

With version 11.0 of YawDyn, we have implemented a dynamic inflow model based on a modification of the theory of Pitt and Peters. This replaces the time delay function used in earlier versions of the program. This model runs significantly faster than the blade element/momentum (BEM) model (EQUIL option) because it does not require iteration at each time step (it must still iterate for a trim solution, however). As this is the first release of this model, we recommend it with the usual caveats that accompany any new software.

The model currently implemented uses three parameters to describe the distribution of the induced velocities over the rotor; the mean (0P), the cosine term which gives the top-to-bottom variation (1P) and the sine term which gives the side-to-side variation (1P). The radial distribution of the induced velocities in the model itself is linear. However, use of Prandtl's tip loss model in conjunction with the dynamic inflow model compensates for this limitation at the blade tips.

As mentioned above, YawDyn must seek a trim solution for the dynamic inflow parameters in addition to the trim solution for blade flap parameters. The dynamic inflow parameters normally trim faster than the blade flap parameters, so the additional calculations are not usually noticeable. However, if you experience trouble with the trim solution, check the ATOLER parameter, which is used for the trim solution tolerance.

The dynamic inflow effect is often insignificant, so results in most cases should not differ from the BEM results. The exception is cases with rapid changes in blade angle-of-attack, where the dynamic inflow effect can be significant.

INDEX

A

B

F

G

H

I

M

N

O

P

Q

R

S

T

U

V

W

Y